ORACLE

涵盖Oracle Database 11*g* R2

Oracle RMAN 11*g* Backup and Recovery

ORACLE DATABASE 11*g*

Oracle Database 11*g* RMAN备份与恢复

(美) Robert G. Freeman
Matthew Hart 著
王念滨 陈子阳 译

ORIGINAL • AUTHENTIC
Oracle Press
ONLY FROM OSBORNE

清华大学出版社

Oracle Database 11*g* RMAN备份与恢复

Oracle RMAN 11*g* Backup and Recovery

ORACLE DATABASE 11*g*

涵盖R2

最大程度地利用Oracle恢复管理器的功能

《Oracle Database 11*g* RMAN备份与恢复》提供了在硬件、软件、操作发生故障时保护数据库的详细信息。详细解释了如何配置数据库，创建精确的归档以及如何执行系统还原。通过命令行或Oracle企业管理器(OEM)使备份进程自动化，执行Oracle 闪回恢复操作以及集成云计算技术。作为权威的资源，《Oracle Database 11*g* RMAN备份与恢复》也提供有关创建报告、优化性能以及执行第三方管理实用程序的相关资料。

主要内容

- 创建、配置、维护Oracle恢复管理器(Oracle RMAN)
- 使用Oracle Secure Backup管理物理和虚拟介质
- 使用Oracle RMAN目录、包和控制文件
- 使用Amazon Web Service云作为一种异地存储解决方案
- 创建联机、脱机以及增量系统备份
- 执行完整或部分Oracle RMAN数据库还原
- 使用Oracle闪回技术修正用户引起的错误
- 在本地或远程服务器上创建数据库副本和备用数据库
- 使用Oracle Real Application Clusters(Oracle实时应用集群)及同步和拆分技术

作者简介

Robert G. Freeman长期从事Oracle DBA的工作。目前为止，他已经撰著过多本书籍，其中包括Oracle出版社出版的*Oracle Database 11g New Features*。

Matthew Hart与他人合著过6本由Oracle出版社出版的书籍，他的主要作有：《Oracle Database 10*g*高可用性实现方案——运用RAC、Flashback和Data Guard技术》、*Oracle Enterprise Manager 10g Grid Control Handbook*以及《Oracle Database 11*g* RMAN备份与恢复》等。

如果需要查找Oracle Press书籍的完整列表，请访问www.OraclePressBooks.com

ORIGINAL • AUTHENTIC Oracle Press ONLY FROM OSBORNE

McGraw-Hill
全球智慧中文化
http://www.mheducation.com

上架建议：数据库/Oracle
读者信箱：wkservice@vip.163.com
投稿信箱：bookservice@263.net

ISBN 978-7-302-24645-9
9 787302 246459 >
定价：78.00元

Oracle Database 11g RMAN 备份与恢复

(美) Robert G. Freeman
Matthew Hart 著

王念滨 陈子阳 译

清华大学出版社

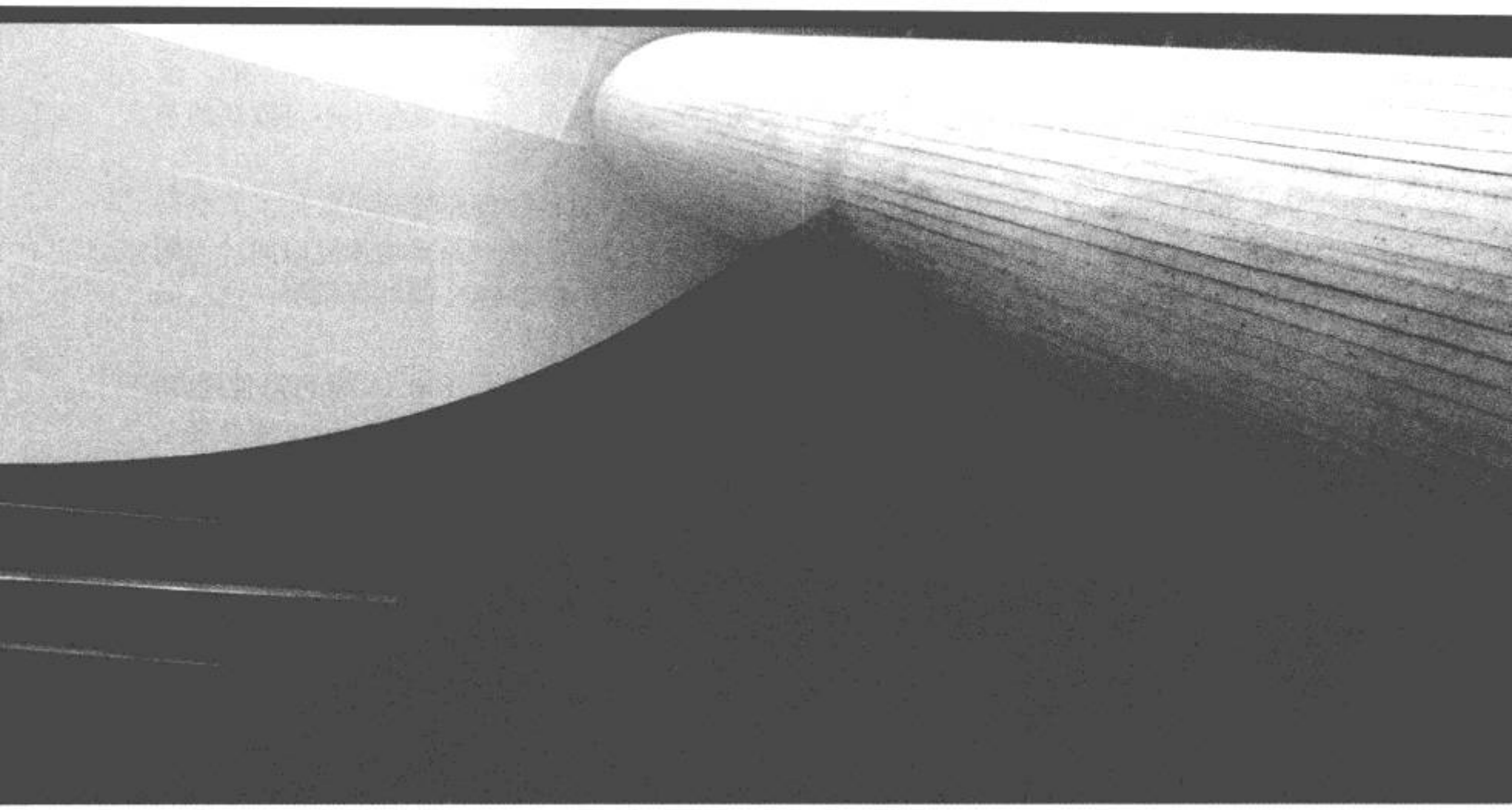
北 京

Robert G. Freeman, Matthew Hart
Oracle RMAN 11g Backup and Recovery
EISBN：978-0-07-162860-0
Copyright © 2010 by The McGraw-Hill Companies, Inc.
All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher
This authorized Chinese translation is jointly published by McGraw-Hill Education (Asia) and Tsinghua University Press. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.
Copyright © 2011 by McGraw-Hill Education (Asia), a division of the Singapore Branch of The McGraw-Hill Companies, Inc. and Tsinghua University Press.
版权所有。未经出版人事先书面许可，对本出版物的任何部分不得以任何方式或途径复制或传播，包括但不限于复印、录制、录音，或通过任何数据库、信息或可检索的系统。
本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和清华大学出版社合作出版。此版本经授权仅限在中华人民共和国境内(不包括香港特别行政区、澳门特别行政区和台湾)销售。
版权©2011 由麦格劳-希尔(亚洲)教育出版公司与清华大学出版社所有。
北京市版权局著作权合同登记号　图字：01-2010-5500
本书封面贴有 McGraw-Hill 公司防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933
图书在版编目(CIP)数据

Oracle Database 11g RMAN 备份与恢复/(美)弗里曼(Freeman, R. G.)，(美)哈特(Hart, M.) 著；王念滨，陈子阳 译. —北京：清华大学出版社，2011.4
书名原文：Oracle RMAN 11g Backup and Recovery
ISBN 978-7-302-24645-9
Ⅰ. O… Ⅱ. ①弗… ②哈… ③王… ④陈… Ⅲ. 关系数据库—数据库管理系统，Oracle 11g Ⅳ. TP311.138
中国版本图书馆 CIP 数据核字(2011)第 006354 号

责任编辑：王　军　梅　丹
装帧设计：孔祥丰
责任校对：胡雁翎
责任印制：何　芊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：清华大学印刷厂
装 订 者：三河市李旗庄少明装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：39.25　**字　数**：907 千字
版　　次：2011 年 4 月第 1 版　　**印　　次**：2011 年 4 月第1 次印刷
印　　数：1～4000
定　　价：78.00 元

产品编号：033606-01



前　言

本书是关于RMAN备份与恢复的最新版本。Oracle Database 11*g*是值得信赖的数据库版本，其RMAN对先前版本进行了改进，增加了一些新的功能和出色的新特性。从Oracle 8版本开始，多年来RMAN不断地进行改进以期获得最佳的效果。

0.1　解决旧问题和提出新问题

在《Oracle 9*i* RMAN备份与恢复技术》(清华大学出版社引进并出版)一书中，我们在前言中提出了如下问题：如何在可用性与可恢复性之间寻求平衡？然后通过全面介绍Oracle的备份和恢复解决方案，我们回答了这个问题。此书的不错销量说明很多人喜欢这个答案。我们在用户开始真正地采用RMAN作为备份与恢复解决方案的时候出版了该书籍。

随着《Oracle Database 10*g* RMAN备份与恢复》(清华大学出版社引进并出版)一书的

问世，我们发现许多读者对 RMAN 的了解更加深入。与此同时，他们提出了一些更加复杂的问题，尝试掌握 Oracle Database 10g 提供的所有新特性。随着数据库不断发展，平均恢复时间成为人们关注的焦点，RMAN 成为数据库管理员常用工具集中不可或缺的重要工具之一。

Oracle Database 11g 发布以来，上述趋势还在延续。目前数据库管理员低估了其复杂性，他们正努力跟上已经出现的变化。从网格计算到高可用性以及平均恢复时间，问题变得越发复杂，解决这些问题也越发困难。当然，我们希望 RMAN 能够提供解决这些问题的方法。

RMAN 是具有 Oracle 数据库许可的。就是说当数据库安装后，RMAN 就可以使用(初始最小配置，本书将会深入讨论这些配置)。RMAN 可以用于备份最大、最小或最复杂等各种类型的 Oracle 数据库。可以备份设置在小型服务器上的单实例数据库，也可以备份设置在多个服务器上的多节点 RAC 集群数据库。Oracle Database 11g RMAN 仍然保留了您所熟悉的各项功能，同时提供了大量的新特性以提高其性能。

0.2 为 DBA 和系统管理员提供帮助

为 Oracle 数据库选择稳定可靠的备份策略时，可能出现的最棘手的问题是：备份策略总是会使数据库管理员和系统管理员的任务出现重叠。制定 RMAN 的备份策略也存在同样的问题。RMAN 的备份策略与 Oracle RDBMS 的集成意味着系统管理员必须具备 Oracle 数据库的实际经验。但是，对外部磁带存储系统和网络拓扑结构的依赖又使得 DBA 管理网络计算机系统的能力显得非常重要。这就引发了职责的分工，对两边来说都是一个难题。

此外，业务需求将模糊 DBA 和系统管理员的职责范围。或者更为准确地说，DBA 会发现他们的工作不断增加，已经扩展到要做一些系统管理员的工作，而系统管理员则会发现要花费更多的时间来学习 SQL 命令。

本书将解决这种角色上的重叠，具体体现在针对数据库备份这个最容易产生角色重叠的领域提供指导性的建议。

0.3 更为优秀的 RMAN

从 Oracle 8.0.3 版本(Oracle 8 的第一个产品版本)开始引入了 RMAN。在这之前，Oracle 提供的直接在磁带上生成数据流备份的接口使用 Export 实用程序进行逻辑备份，或使用 Enterprise Backup Utility(EBU)。现在已经不再使用 EBU，这也是最后一次提及该工具。

最初的 RMAN 版本难免有不尽人意之处，而且存在不少错误。但是，其后的版本添加了一些新的功能，修正了软件故障，并且改进了接口。理解 RMAN 版本演化的最好方法是想象一下演示人类进化过程的传统海报。在海报上的左边，是一只靠四肢行走的猴

子。稍往右看，是一个逐渐直立行走的人，直到最右边，所看到的是一个完全直立行走的现代人。

在 Oracle 9*i* 版本中，RMAN 的功能更加完备，并且已经成为用于高可用性数据库系统的各种重要策略的必要组件。

现在，RMAN 已经经历了两个 10*g* 版本，在功能上不断地改进和完善，其目标是成为功能完善的可用组件。

0.4　本书主要内容

本书介绍了 Oracle Database 11*g* Release 2 的最新功能，帮助您充分了解 RMAN 接口的最新功能，并且说明了所增加的新特性。所有代码示例和体系结构说明都基于 RMAN 的 11*g* R2 版本。

如果您使用的仍然是 Oracle 和 RMAW 早期的版本(Oracle 8*i*, Oracle 9*i*, Oracle Database 10*g*)，该书仍然对您有帮助，尽管一些新特性您体验不到。显而易见的是，本书全面介绍了在 Oracle 11*g* 中执行各项操作的方式，但是没有指出或引用前面版本《Oracle 9*i* RMAN 备份与恢复》或者《Oracle Database 10*g* RMAN 备份与恢复》的内容。

0.5　更有效地使用本书

与其他技术指南一样，可以通过阅读本书来熟悉 RMAN 及其在高可用性或灾难恢复解决方案中的作用。本书采用由简入繁、循序渐进的方式，逐步地讨论了从基本原则到设置到备份的内容，最后介绍了高级功能与各种实习。

第 I 部分介绍了 Oracle RDBMS 中备份与恢复的基本原理，以及 RMAN 的概念和体系结构。该部分的两章内容是其他章节的基础，我们建议您认真阅读和理解这些概念。如果能够理解该部分中描述的概念和内部运行机制，就能轻而易举地理解本书的其余部分。

第 II 部分讨论了 RMAN 的初始设置。首先介绍了所有可能的 RMAN 配置选项，然后讨论了 RMAN 与介质管理器的集成。介质管理层允许用户将备份直接写入磁带设备。目前市场上有多种介质管理产品，该部分将讨论 4 种最常用的产品：Oracle 自带的 Secure Backup、VERITAS NetBackup、EMC NetWorker Module for Oracle，以及 IBM Tivoli Storage Manager。

第III部分提供了有关 RMAN 用法的基础知识，其范围从最基本的备份操作到最高级的恢复选项。我们将讨论目录维护和查看目录的方法，以便更有效地管理不断存储的备份。此处将介绍 Oracle 中经过重新设计的 Enterprise Manager 产品，并将介绍如何使用闪回技术从逻辑错误中恢复。最后，还将讨论为优化性能对 RMAN 备份与恢复进行的调整。

第IV部分介绍了备份与恢复操作外的更高级的功能，让您了解利用 RMAN 完成简单备份操作之外能完成的其他任务。该部分的内容包括：如果使用 RMAN 备份生成数据库

的克隆副本和如何使用备份创建 Oracle Data Guard 使用的备用数据库；根据特殊的需求和要求在 Real Application Clusters(RAC)环境中使用 RMAN；对一系列 RMAN 案例的分析深入研究了需要使用 RMAN 的各种情况。

第 V 部分中的附录 A 包括 RMAN 语法参考，可以借助该附录成功地构建 RMAN 命令；附录 B 探讨了 RMAN 目录，以及数据库中的 v$视图和恢复目录中的 rc_*视图；附录 C 详细介绍了如何设置 RMAN 测试环境。在实现产品备份与恢复策略前，您可以使用这个测试环境来测试各种 RMAN 功能以达到用最少的工作高效率地将该书投入实践操作。

0.6 RMAN 实践

当然，您可能不会细读书中的所有内容。一本优秀的技术书籍应该是摆放在计算机旁边，带有折角页、重点章节标记和书签标记，供您随时查阅。

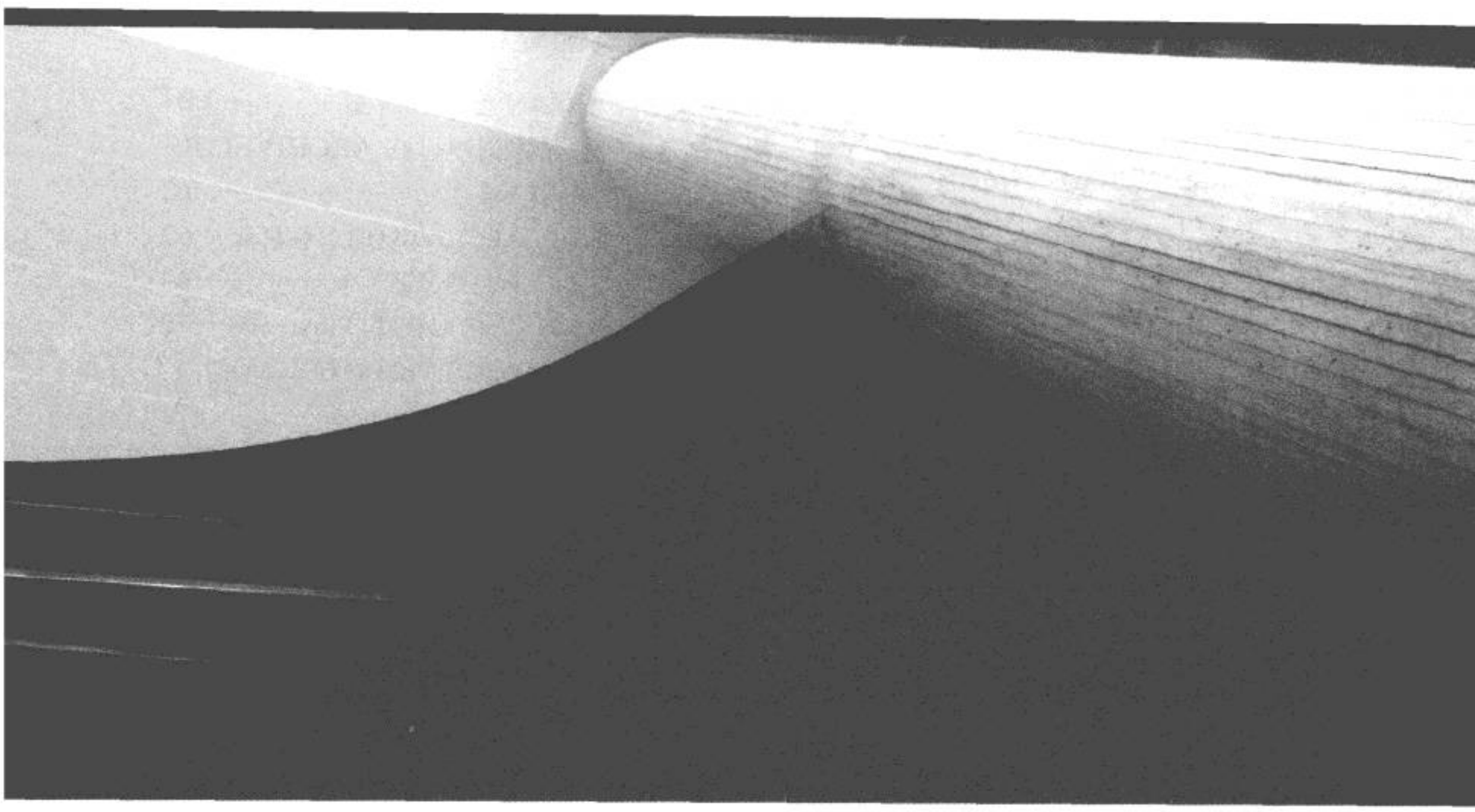
除了概念解释外，本书实际上是一本参考指南。我们在书中综合了各种有用的技术和简便的方法。即使不熟悉 RMAN 的整个体系结构，也可以随时运用这些方法。有时候，尤其是在备份与恢复操作中，我们只需要知道该怎么做，而不需要知道为什么要这样做。因为谁都不希望被一大堆恢复工作和操作语法弄得焦头烂额，而使得产品数据库处于空闲状态，并导致时间和金钱的白白流失。

因此，为了突出重点，我们在本书中引入了一些 RMAN 实习的章节，阅读过 Oracle 先前版本的读者应该已经非常熟悉这些 RMAN 实习。在这些 RMAN 实习中，我们提供了用于执行指定操作的代码或用于完成特定项目的一系列步骤，以帮助您快速完成各种工作。我们为 RMAN 实习添加了灰色方框，当您看到这种方框时，就可以知道下面的页面将介绍一些实际的步骤，遵循这些步骤可以快速地完成工作。可以将 RMAN 实习视为食谱，它提供了快速做好一顿饭所需的各种配料和混合这些配料的说明。

我们再次建议您逐章地阅读本书。因为理解产品的概念是最重要的，特别是用于保护最有价值的数据库的产品。

总之，RMAN 是一个值得深入研究和充分利用的产品，它能够节省您的时间和精力，并帮助您解决运行中断压力等各种问题。

最后，请将您对本书的建议发送至 wkservice@vip.163.com，我们将不胜感激。



目　录

第Ⅰ部分　Oracle RMAN 11*g* 基础知识

第 1 章　Oracle Database 11*g* 备份与恢复体系结构概述 …… 3

1.1　备份和恢复概要 …… 4

1.1.1　高可用性 …… 5

1.1.2　备份和恢复 …… 5

1.2　需要掌握的 Oracle 术语 …… 7

1.3　数据库软件的控制 …… 9

1.4　Oracle 体系结构 …… 11

1.5　Oracle 内存和 RMAN …… 12

1.6　更多关于 Oracle 重做日志的内容 …… 16

1.6.1　ARCHIVELOG 模式与 NOARCHIVELOG 模式 …… 20

1.6.2　Oracle 逻辑结构 …… 20

1.7　组合情况 …… 20

1.7.1　数据库的启动和关闭 …… 21

1.7.2　使用数据库和内核 …… 23

1.8　Oracle 备份与恢复入门 …… 25

1.8.1　逻辑备份与恢复 …… 25

1.8.2　Oracle 物理备份与恢复 …… 25

1.8.3　备份其他 Oracle 组件 …… 30

1.9　小结 …… 31

第 2 章 RMAN 体系结构概述 …… 33
2.1 服务器管理恢复 …… 34
2.2 RMAN 实用程序 …… 34
2.3 RMAN 备份的网络拓扑结构 …… 36
2.3.1 远程运行 RMAN …… 36
2.3.2 在目标数据库的 ORACLE_HOME 中本地运行 RMAN …… 38
2.4 数据库控制文件 …… 39
2.4.1 在控制文件中重用记录 …… 39
2.4.2 快照控制文件 …… 41
2.5 RMAN 服务器进程 …… 42
2.5.1 RMAN 信道进程 …… 42
2.5.2 RMAN 和 I/O 从属 …… 43
2.6 RMAN 使用的 SYS 数据包 …… 43
2.6.1 SYS.DBMS_RCVMAN 数据包 …… 43
2.6.2 SYS.DBMS_BACKUP_RESTORE 数据包 …… 43
2.7 备份数据块 …… 44
2.7.1 数据块备份概述 …… 44
2.7.2 块级备份的优点 …… 45
2.8 内存中的 RMAN …… 47
2.8.1 输入内存缓冲区 …… 47
2.8.2 用于还原的内存缓冲区 …… 48
2.8.3 RMAN 内存利用：PGA 与 SGA …… 49
2.9 恢复目录 …… 50
2.10 辅助数据库 …… 51
2.11 版本兼容性 …… 52
2.11.1 目标数据库与 RMAN 可执行文件 …… 53
2.11.2 恢复目录数据库与恢复目录模式 …… 53
2.11.3 辅助数据库 …… 53
2.12 RMAN 进程 …… 53
2.13 闪回恢复区 …… 56
2.14 小结 …… 57

第 II 部分 设置原则和实践

第 3 章 RMAN 设置和配置 …… 61
3.1 配置数据库以在 ARCHIVELOG 模式中运行 …… 62
3.1.1 ARCHIVELOG 目标目录 …… 62
3.1.2 闪回恢复区 …… 64
3.1.3 是否应该使用 FRA …… 70
3.1.4 ARCHIVELOG 模式之间的切换 …… 71
3.1.5 使用 Oracle 数据库配置助手创建数据库 …… 71
3.2 Oracle Database 11g 故障诊断基础架构 …… 73
3.3 RMAN 命令行 …… 75
3.3.1 通过 RMAN 命令行连接 …… 75
3.3.2 RMAN 客户端兼容性 …… 78
3.3.3 使用 RMAN 的 connect 命令 …… 78
3.3.4 退出 RMAN 客户端 …… 79
3.4 为 RMAN 操作配置数据库 …… 79
3.4.1 设置数据库用户 …… 79
3.4.2 设置数据库安全性 …… 80
3.4.3 设置 CONTROL_FILE_RECORD_KEEP_TIME 参数 …… 81
3.5 配置 RMAN 默认设置 …… 82
3.5.1 configure 命令介绍 …… 82
3.5.2 配置多种 RMAN 默认设置 …… 83
3.5.3 configure 命令的使用示例 …… 84
3.5.4 使用共享服务器的情况 …… 95
3.6 RMAN 配置任务总结 …… 95
3.7 其他备份和恢复设置与配置问题 …… 96
3.8 小结 …… 96

第 4 章 介质管理问题 …… 97
4.1 使用磁带备份的原因 …… 98
4.2 RMAN 和介质管理器概述 …… 99
4.2.1 介质管理器目录 …… 99

4.2.2 介质管理器的其他软件组件……100
4.2.3 介质管理库(MML)……100
4.2.4 连接 MML 的接口……102
4.3 SBT API……103
4.4 磁带备份的全过程……104
4.5 磁带还原的全过程……104
4.6 使用 sbttest 和 loadsbt.exe……105
4.7 介质管理错误……106
4.8 小结……107

第 5 章 Oracle 安全备份……109
5.1 OSB 的特性……110
5.1.1 OSB 和恢复管理器……110
5.1.2 OSB 和 OSB Express 的区别……111
5.1.3 备份加密……111
5.1.4 快速数据库备份压缩……111
5.1.5 Oracle 安全备份云模块……111
5.2 OSB 的接口……111
5.3 OSB 的组件……112
5.3.1 OSB 守护进程(Daemons)……114
5.3.2 主机访问模式……114
5.3.3 管理数据……114
5.3.4 OSB 用户和类……115
5.3.5 操作系统账户……115
5.3.6 NDMP 主机……116
5.3.7 OSB 权限和类……116
5.4 安装与配置 OSB……117
5.5 使用 OSB 备份 Oracle 数据库及文件系统数据……128
5.6 使用 OSB 云模块备份数据库……133
5.7 小结……136

第 6 章 使用 OSB 云模块将数据库备份到 Amazon Web 服务…… 137
6.1 传统备份：前提与限制……138
6.2 OSB 云模块……138
6.3 云计算的含义……138
6.4 Oracle 与 Amazon 云……139
6.4.1 弹性计算云(EC2)与弹性块存储(EBS)……139
6.4.2 简单存储服务(S3)——Oracle 的云备份解决方案……139
6.4.3 RMAN 备份到 S3：OSB 云模块……139
6.4.4 通过 Internet 备份或 Amazon EC2 备份到 S3……139
6.5 Oracle 云备份的优点……140
6.6 使用 OSB 云模块执行备份……142
6.6.1 列出存储在 S3 中的 RMAN 备份以及备份集……144
6.6.2 通过 Internet 使用 OSB 云模块和 Amazon S3 优化备份与恢复……144
6.7 许可问题……146
6.8 小结……146

第 7 章 使用 Oracle 的 VERITAS NetBackup 增强 RMAN…… 147
7.1 关键特性……148
7.2 必要的组件……148
7.3 存储/介质设备配置……149
7.4 NetBackup 的安装……149
7.4.1 安装 NetBackup for Oracle 代理之前的任务……150
7.4.2 NetBackup for Oracle 代理安装步骤……150
7.5 链接 Oracle 与 NetBackup 介质管理器的方法……151
7.5.1 自动链接方法……151
7.5.2 手动链接方法……152
7.6 体系结构……153
7.7 配置 NetBackup 策略……154
7.7.1 添加新策略……154

7.7.2 定义时间表 …… 157
7.7.3 定义备份选项 …… 159
7.7.4 定义策略客户端 …… 159
7.8 管理到期的备份映像 …… 160
7.8.1 使用 NetBackup 存储库删除到期的备份 …… 160
7.8.2 使用 RMAN 删除到期的备份 …… 160
7.9 RMAN 样本脚本 …… 161
7.10 故障诊断 …… 162
7.10.1 使用 NetBackup 日志 …… 162
7.10.2 确定正在使用的库 …… 163
7.11 维护安全的最佳方法 …… 163
7.12 成本调整 …… 163
7.13 小结 …… 164

第 8 章 配置 Oracle 的 HP Data Protector …… 165
8.1 集成 Oracle 与 DataProtector …… 166
8.1.1 支持矩阵 …… 166
8.1.2 集成组件 …… 166
8.2 在 Data Protector 上配置 RMAN 备份 …… 170
8.3 RMAN 备份 …… 174
8.3.1 备份方法 …… 174
8.3.2 备份过程 …… 175
8.4 使用 Data Protector GUI 还原 Oracle 数据库 …… 175
8.4.1 还原控制文件 …… 176
8.4.2 还原 Oracle 数据库对象 …… 176
8.5 Oracle RMAN 元数据与 Data Protector 介质管理数据库同步 …… 177
8.6 小结 …… 178

第 9 章 RMAN 与 Tivoli Storage Manager …… 179
9.1 Tivoli Storage Manager 概述 …… 180
9.1.1 TSM 服务器系统对象 …… 181
9.1.2 TSM 客户端 …… 182
9.1.3 TSM Administration Center 和 Web 客户端 …… 183
9.2 使用 TDPO 完成 RMAN 备份 …… 187
9.3 删除数据库备份 …… 192
9.4 公共备份场景的故障诊断 …… 193
9.5 其他故障诊断 …… 194
9.6 小结 …… 195

第 10 章 恢复目录的使用 …… 197
10.1 恢复目录的含义 …… 198
10.1.1 创建恢复目录 …… 199
10.1.2 使用虚拟私有目录 …… 202
10.2 合并多个恢复目录 …… 203
10.3 维护恢复目录 …… 204
10.3.1 取消 RMAN 中的数据库注册 …… 204
10.3.2 数据库迁移/升级问题 …… 205
10.3.3 手动重新设置数据库对应物(reset catalog) …… 205
10.3.4 手动执行恢复目录的再同步操作(resync catalog) …… 205
10.3.5 清除恢复目录记录 …… 206
10.4 备份恢复目录 …… 206
10.5 恢复目录视图 …… 206
10.5.1 RC_ARCHIVED_LOG (V$ARCHIVED_LOG) …… 207
10.5.2 RC_BACKUP_CONTROLFILE (V$BACKUP_DATAFILE) …… 207
10.5.3 RC_BACKUP_CORRUPTION (V$BACKUP_CORRUPTION) …… 208
10.5.4 RC_BACKUP_DATAFILE (V$BACKUP_DATAFILE) …… 208

10.5.5 RC_BACKUP_FILES (V$BACKUP_FILES)···· 208
10.5.6 RC_BACKUP_PIECE (V$BACKUP_PIECE) ··· 209
10.5.7 RC_BACKUP_REDOLOG (V$BACKUP_ REDOLOG)····· 209
10.5.8 RC_BACKUP_SET (V$BACKUP_SET) ····· 209
10.5.9 RC_BACKUP_SPFILE (V$BACKUP_ SPFILE) ····· 209
10.5.10 RC_CONTROLFILE_ COPY(V$DATAFILE_ COPY)····· 209
10.5.11 RC_COPY_CORRUPTION (V$COPY_ CORRUPTION) ····· 209
10.5.12 RC_DATABASE (V$DATABASE)····· 209
10.5.13 RC_DATABASE_BLOCK_ CORRUPTION (V$DATABASE_BLOCK_ CORRUPTION) ····· 210
10.5.14 RC_DATABASE_ INCARNATION (V$DATABASE_ INCARNATION) ····· 210
10.5.15 RC_DATAFILE (V$DATAFILE)····· 210
10.5.16 RC_DATAFILE_COPY (V$DATAFILE_ COPY)····· 210
10.5.17 RC_LOG_HISTORY (V$LOG_HISTORY) ··· 210
10.5.18 RC_OFFLINE_RANGE (V$OFFLINE_ RANGE)·····211
10.5.19 RC_REDO_LOG (VLOG, VLOGFILE)·····211
10.5.20 RC_REDO_THREAD (V$THREAD)·····211
10.5.21 RC_RESYNC·····211
10.5.22 RC_RMAN_ CONFIGURATION (V$RMAN_ CONFIGURATION) ····211
10.5.23 RC_TABLESPACE (V$TABLESPACE)·····212
10.5.24 RC_TEMPFILE (V$TEMPFILE)·····212
10.5.25 为 Oracle 企业管理器提供的目录视图·····212
10.6 小结 ·····213

第 11 章 RMAN 备份·····215
11.1 RMAN 备份优于脚本化备份 ·····215
11.2 RMAN 兼容性问题·····217
11.3 监控 RMAN 备份状态·····218
11.4 脱机 RMAN 数据库备份·····218
11.4.1 使用默认的设置执行脱机备份操作·····219
11.4.2 不使用默认的设置执行脱机备份操作·····222
11.5 Backup 命令选项 ·····225
11.5.1 多段备份 ·····225
11.5.2 压缩 ·····226
11.5.3 标记和还原点·····227
11.5.4 限定备份的影响·····228
11.5.5 限定备份集的大小·····228
11.5.6 备份到特定的设备类型····229
11.5.7 修改备份集的保留策略····229
11.5.8 归档日志删除策略·····231

11.5.9 重写 configure exclude 命令 ····· 232
11.5.10 使用 backup 命令检查数据库的错误 ····· 232
11.5.11 跳过脱机的、不可访问的或只读的数据文件 ····· 232
11.5.12 强制备份只读的数据文件 ····· 233
11.5.13 基于上次备份时间来备份数据文件 ····· 233
11.5.14 在 RMAN 复制设备上生成备份副本 ····· 235
11.5.15 捕获隐蔽的控制文件 ····· 235
11.6 set 命令介绍 ····· 235
11.7 联机 RMAN 数据库备份 ····· 236
11.7.1 联机数据库备份 ····· 236
11.7.2 表空间备份 ····· 239
11.7.3 数据文件备份 ····· 239
11.7.4 归档重做日志备份 ····· 240
11.7.5 控制文件和服务器参数文件备份 ····· 241
11.7.6 备份集备份 ····· 241
11.7.7 闪回恢复区备份 ····· 242
11.8 副本 ····· 242
11.8.1 映像副本介绍 ····· 242
11.8.2 数据库、表空间和数据文件的映像副本 ····· 242
11.8.3 控制文件副本 ····· 243
11.8.4 ARCHIVELOG 映像副本 ····· 244
11.9 增量 RMAN 备份 ····· 244
11.9.1 块更改跟踪文件 ····· 245
11.9.2 基本备份 ····· 246
11.9.3 差异增量备份与累积增量备份 ····· 246
11.10 备份操作练习 ····· 250
11.11 小结 ····· 253

第 12 章 RMAN 还原与恢复 ····· 255
12.1 RMAN 还原与恢复基础 ····· 256
12.2 还原数据库前的准备工作 ····· 257
12.3 RMAN 还原与恢复前的操作 ····· 257
12.3.1 有关恢复、恢复目录及 MML 层的注意事项 ····· 258
12.3.2 还原服务器参数文件 (SPFILE) ····· 258
12.3.3 还原控制文件 ····· 263
12.4 restore 和 recover 命令 ····· 270
12.4.1 restore 命令 ····· 270
12.4.2 recover 命令 ····· 271
12.5 在 NOARCHIVELOG 模式中还原和恢复数据库 ····· 271
12.5.1 还原的准备工作 ····· 271
12.5.2 在不同的位置上还原数据库 ····· 273
12.6 在 ARCHIVELOG 模式中恢复数据库 ····· 276
12.6.1 故障点数据库恢复 ····· 277
12.6.2 表空间恢复 ····· 281
12.6.3 数据文件恢复 ····· 282
12.6.4 使用增量备份的恢复操作 ····· 282
12.7 联机重做日志丢失的恢复 ····· 283
12.7.1 非活动的联机重做日志文件组成员的丢失 ····· 284
12.7.2 非活动的联机重做日志文件组的丢失 ····· 284
12.7.3 丢失活动但非当前状态的联机重做日志文件组 ····· 285
12.7.4 丢失当前联机重做日志文件组 ····· 286
12.8 数据恢复顾问 ····· 286
12.9 小结 ····· 292

第III部分 更有效地使用 RMAN

第 13 章 使用 Oracle 企业管理器实现备份和恢复操作 …… 295

13.1 Oracle 企业管理器：新范例 …… 296

13.2 Grid Control …… 298

13.2.1 Grid Control 的体系结构 …… 299

13.2.2 安装和配置 Grid Control …… 300

13.3 Database Control …… 300

13.3.1 Database Control 的体系结构 …… 301

13.3.2 安装和配置 Database Control …… 302

13.4 在企业管理器中配置备份设置 …… 305

13.4.1 设备配置 …… 306

13.4.2 备份集配置 …… 307

13.4.3 策略配置 …… 308

13.4.4 在 OEM 备份配置中无法配置的内容 …… 308

13.5 配置恢复设置 …… 310

13.5.1 实例恢复 …… 310

13.5.2 介质恢复 …… 310

13.5.3 闪回恢复 …… 311

13.6 在 OEM 中配置恢复目录 …… 312

13.7 企业管理器中的数据库备份 …… 314

13.7.1 Oracle 建议备份策略 …… 314

13.7.2 调度自定义的备份 …… 317

13.7.3 RMAN 脚本作业与调度备份向导 …… 318

13.8 在企业管理器中执行恢复操作 …… 320

13.8.1 数据恢复顾问和 OEM 检查器 …… 321

13.8.2 用户直接恢复 …… 325

13.9 备份管理和报告 …… 328

13.9.1 管理当前备份 …… 328

13.9.2 管理还原点 …… 329

13.9.3 创建备份报告 …… 329

13.10 在企业管理器中复制数据库 …… 329

13.11 小结 …… 331

第 14 章 RMAN 的高级恢复主题 …… 333

14.1 不完全恢复 …… 333

14.1.1 使用 resetlogs 命令 …… 334

14.1.2 创建恢复点 …… 335

14.1.3 基于时间的恢复 …… 336

14.1.4 基于 SCN 的恢复 …… 336

14.1.5 基于日志序列的恢复 …… 336

14.1.6 基于删除的恢复 …… 337

14.1.7 使用还原点恢复 …… 337

14.2 其他 RMAN 恢复主题 …… 337

14.2.1 只读表空间的恢复 …… 337

14.2.2 归档重做日志的还原 …… 338

14.2.3 数据文件副本的还原 …… 338

14.2.4 恢复损坏的数据块 …… 338

14.2.5 恢复前一个对应物 …… 341

14.3 表空间时间点恢复 …… 344

14.3.1 执行自动的 TSPITR …… 345

14.3.2 手动执行 TSPITR …… 348

14.3.3 TSPITR 限制 …… 353

14.4 验证备份可恢复 …… 354

14.4.1 restore preview 命令 …… 354

14.4.2 使用 validate 和 check logical 参数的 restore 命令 …… 356

14.4.3 使用 validate backupset 命令 …… 358

14.5 跨平台的数据库移动和 RMAN …… 359

14.5.1 可跨平台传送的表空间……359
14.5.2 字节排序和数据文件转换……360
14.5.3 跨平台移动数据库……361
14.6 处理 RMAN 的异常情况……362
14.7 小结……363

第 15 章 从用户错误中恢复：闪回技术……365
15.1 为不可避免的问题做准备：闪回技术……366
15.2 闪回查询……366
15.2.1 闪回和撤消段简介……366
15.2.2 执行闪回查询……367
15.2.3 使用 Oracle 企业管理器执行闪回版本查询……368
15.3 闪回表……371
15.3.1 从 SQL 中执行闪回表操作……372
15.3.2 在 Oracle 企业管理器中使用闪回表……372
15.4 闪回事务……374
15.5 闪回删除……376
15.6 闪回数据库……380
15.6.1 闪回日志……380
15.6.2 闪回保存目标……381
15.6.3 闪回数据库的调整……382
15.7 闪回数据归档(完全撤消)……384
15.8 小结……385

第 16 章 RMAN 维护……387
16.1 RMAN 的维护……388
16.1.1 交叉校验 RMAN 备份……388
16.1.2 RMAN 备份的验证……392
16.1.3 备份保留策略……393
16.1.4 change 命令……396
16.1.5 delete 命令……404
16.1.6 在 RMAN 中为其他备份编写目录……405
16.2 RMAN 存储的脚本……406
16.2.1 创建存储的脚本……407
16.2.2 查询存储脚本信息的恢复目录……407
16.2.3 修改存储的脚本……407
16.2.4 删除存储的脚本……407
16.2.5 使用存储的脚本……407
16.2.6 打印存储的脚本……408
16.3 删除数据库……409
16.4 小结……409

第 17 章 RMAN 的监控和报告……411
17.1 RMAN 的 list 命令……411
17.1.1 列出对应物……412
17.1.2 列出备份……412
17.1.3 列出映像副本……423
17.2 RMAN 的 report 命令……426
17.2.1 报告最近没有被备份的数据文件……426
17.2.2 报告备份冗余或恢复窗口……427
17.2.3 报告数据文件的不可恢复操作……428
17.2.4 报告数据库模式……428
17.2.5 报告过时的备份……428
17.3 用于报告的数据字典视图……430
17.4 小结……431

第 18 章 调整 RMAN 备份与恢复操作的性能……433
18.1 调整 RMAN 前的工作……434
18.1.1 可以达到的 RMAN 性能……434
18.1.2 使用合适的硬件……434
18.1.3 调整数据库……436
18.2 调整 RMAN……438
18.2.1 调整 RMAN 设置……439
18.2.2 调整 MML 层……441

18.2.3 确认与数据库相关的RMAN 问题 ····· 442
18.3 跟踪 RMAN 会话 ····· 448
18.4 小结 ····· 450

第Ⅳ部分 Oracle 生态系统中的 RMAN

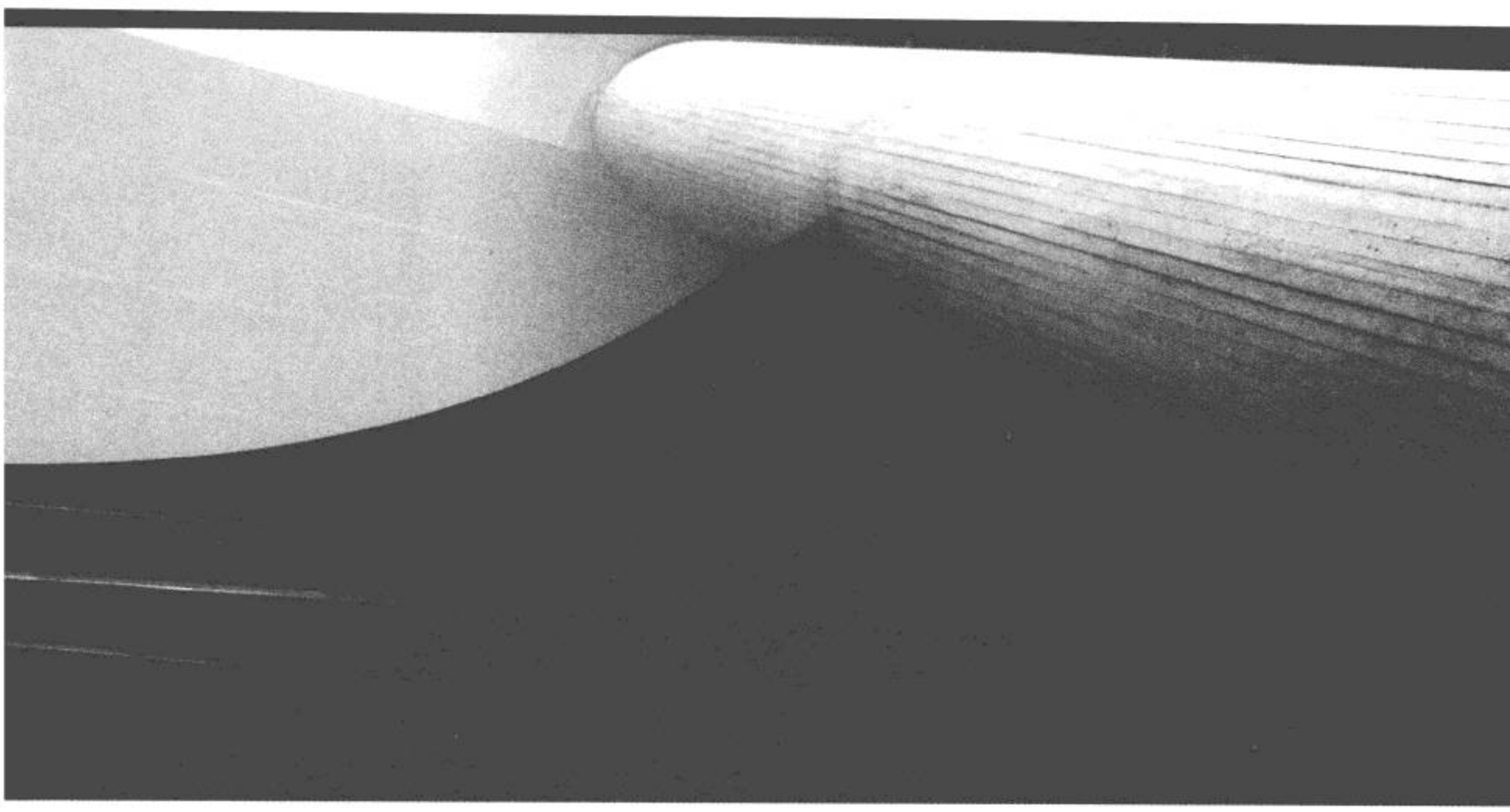
第 19 章 复制目标数据库 ····· 453
19.1 RMAN 复制的基础 ····· 454
19.1.1 使用 RMAN 复制的原因 ····· 454
19.1.2 RMAN 复制的不同类型 ····· 455
19.1.3 复制体系结构 ····· 456
19.2 复制的位置 ····· 461
19.2.1 复制到相同的服务器 ····· 461
19.2.2 复制到相同服务器的不同 ORACLE_HOME 目录 ····· 463
19.2.3 复制到远程服务器 ····· 463
19.2.4 复制和网络 ····· 466
19.3 在相同服务器上复制 ····· 469
19.4 在远程服务器上复制 ····· 472
19.5 11g 的无目标复制 ····· 474
19.6 小结 ····· 476

第 20 章 RMAN 与 Data Guard ····· 477
20.1 RMAN 与备用数据库 ····· 478
20.1.1 使用 RMAN 创建备用数据库的必要条件 ····· 478
20.1.2 duplicate…for standby 命令 ····· 479
20.2 从备用数据库生成备份 ····· 483
20.2.1 从备用数据库生成的数据文件备份 ····· 485
20.2.2 从备用数据库备份归档日志 ····· 485
20.2.3 使用闪回数据库重新实例化备用数据库 ····· 486
20.3 小结 ····· 486

第 21 章 RMAN 与 Real Application Clusters ····· 487
21.1 Real Application Clusters：独特的备份问题 ····· 488
21.1.1 数据文件备份 ····· 489
21.1.2 归档日志备份 ····· 490
21.2 RAC 恢复问题 ····· 492
21.2.1 还原操作 ····· 493
21.2.2 还原操作期间的介质管理问题 ····· 493
21.2.3 还原操作后的恢复问题 ····· 494
21.3 高级的 RMAN/RAC 功能 ····· 495
21.3.1 复制到单节点系统 ····· 495
21.3.2 单节点上的备用数据库 ····· 498
21.3.3 从单节点备用数据库备份多节点 RAC 数据库 ····· 501
21.4 小结 ····· 501

第 22 章 同步和拆分技术中的 RMAN ····· 503
22.1 同步和拆分：不完整的镜像备份 ····· 504
22.2 同步和拆分卷上的 Oracle 数据库 ····· 506
22.2.1 数据文件 ····· 507
22.2.2 控制文件 ····· 507
22.2.3 重做日志文件 ····· 508
22.2.4 归档日志 ····· 508
22.3 拆分镜像备份的优点 ····· 509
22.3.1 快速时间点恢复 ····· 509
22.3.2 相对快速的备份 ····· 509
22.3.3 在另一个服务器上加载拆分镜像 ····· 509

22.3.4 从拆分镜像中获得备份……510
22.4 RMAN 与同步和拆分……510
22.4.1 使用 RMAN 注册拆分镜像副本……510
22.4.2 从拆分镜像中获得 RMAN 备份……510
22.4.3 廉价获得同步和拆分功能……512
22.4.4 将备用数据库、闪回数据库以及增量应用于同步和拆分……512
22.4.5 Oracle 同步和拆分解决方案的优点……514
22.5 应用于 Windows 的 Oracle 集成卷影复制服务……514
22.6 小结……515
第 23 章 RMAN 案例研究……517
23.1 恢复操作前的准备工作……518
23.1.1 故障的本质……518
23.1.2 可用的恢复选项……518
23.1.3 可能需要的 Oracle 支持……519
23.1.4 在恢复操作期间寻求帮助……519
23.2 恢复案例研究……519
23.3 小结……541
第Ⅴ部分 附 录
附录 A RMAN 语法参考指南……545
附录 B RMAN 脚本示例……603
附录 C 创建 RMAN 测试环境……607



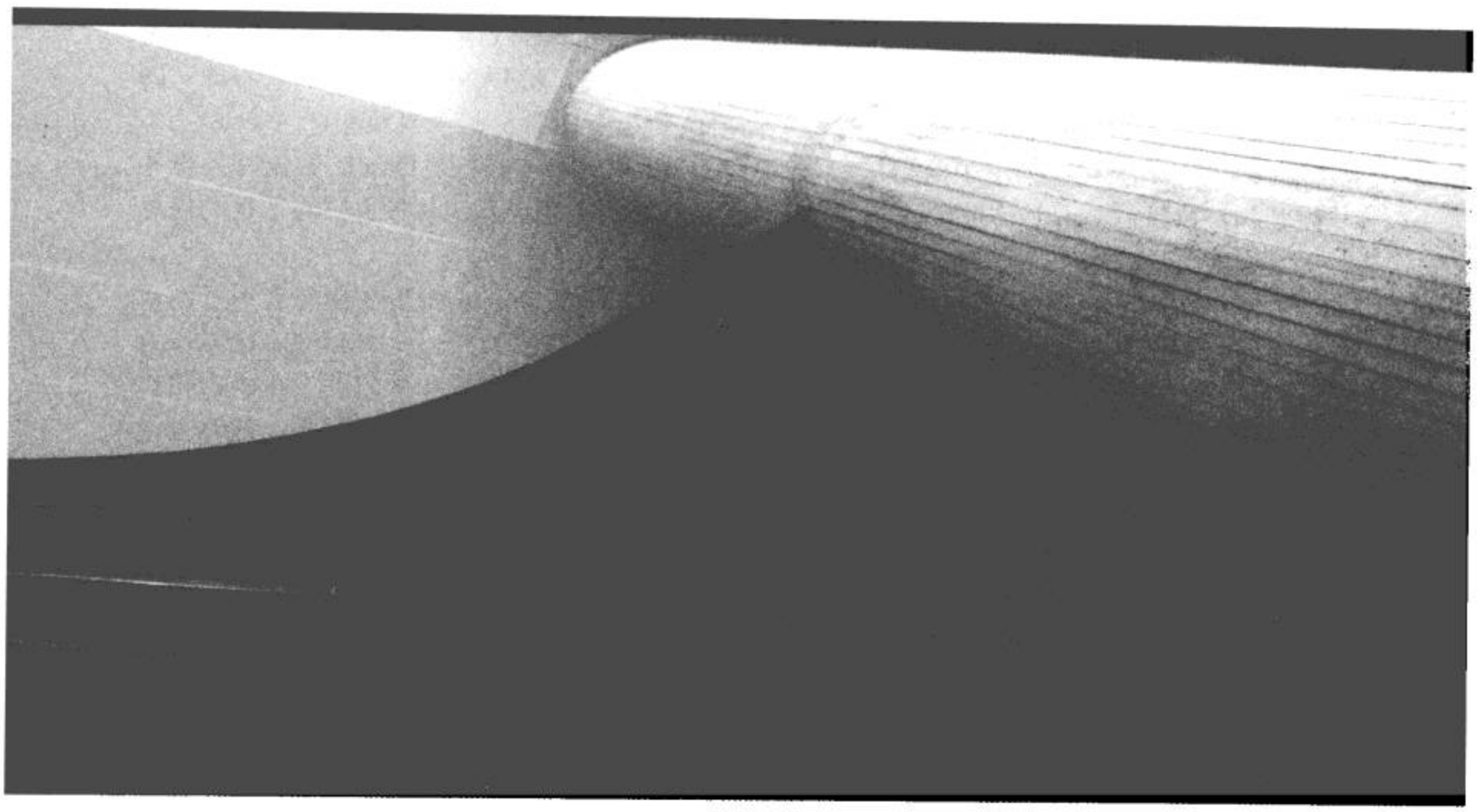
第 I 部分

Oracle RMAN 11*g*

基础知识

第 1 章　Oracle Database 11*g* 备份与恢复体系结构概述

第 2 章　RMAN 体系结构概述



第1章

Oracle Database 11*g*备份与恢复体系结构概述

欢迎阅读《Oracle Database 11*g* RMAN 备份与恢复》一书。如果已经购买了前一版的书籍《Oracle Database 10*g* RMAN 备份与恢复》，您已大概了解本书将要介绍的内容。然而，本书不是简单的修订版。Oracle RMAN 11*g* 具有许多新特性，因此本书除了修订外还添加了许多新内容。希望您发现这些新的特征和修订非常有用。我们也倾听了读者的反馈并针对一些反馈进行了修改。

如果已经使用过 RMAN，您可能会关注 Oracle Database 11g 中的改动对您已有的备份和恢复策略是否存在负面影响，请不要为此感到担心。RMAN 是完全向后兼容的，因此在改为使用 Oracle Database 11g 时，不需要改变已有的备份和恢复策略。

如果您是刚开始使用 RMAN，那么欢迎加入！RMAN 是进行 Oracle 数据库备份和恢复的极好选择。本书将提供成功使用 RMAN 所需的所有信息。本书的目的是帮助您尽快掌握 RMAN 的用法。

然而，在开始深入学习 RMAN 之前，应当对基础的 Oracle 备份和恢复有所了解，这在 Oracle Database 11g 中并没有多大的改动。真正的改动存在于 RMAN，在 Oracle Database 11g 中这样的改动有很多。因此，对于熟悉 Oracle 备份和恢复的人来说，可以阅读本章以加深印象，或者直接从第 2 章开始学习。对于 Oracle 初学者来说，这部分内容会帮助您更好地学习 RMAN，而在随后的章节中将更详细地介绍这些内容。现在，我们就开始学习 Oracle 备份和恢复。

在本章对 Oracle 数据库备份和恢复体系结构的概述中，会介绍下面这些知识：

- 备份和恢复概要
- 需要掌握的一些 Oracle 术语
- Oracle 数据库物理体系结构
- Oracle 操作内核
- ARCHIVELOG 模式操作与 NOARCHIVELOG 模式操作
- Oracle 恢复模式
- Oracle 中的手动备份操作
- Oracle 中的手动恢复操作

在阅读本章的 Oracle 概述的过程中，您会发现掌握 Oracle 产品的工作原理是非常重要的，只有这样才能在恢复数据库时恰当地应用本书所介绍的各种技术。此外，您还会发现备份和恢复数据库并不只是输入一些命令和在磁带驱动器中插入磁带。

错误应用某种技术或者不理解某种体系结构原则的直接后果可能会导致大范围的停机，甚至会丢失数据。实际上，我们在遇到一个恢复实例时花费十小时，但当我们对情况进行研讨后，发现实际上该项工作可能仅需两小时就能完成。在此实例中的时间开销的差别就在于 DBA 缺乏对其需要恢复的数据库的一些基本理解，该问题导致其对恢复方式的选择存在巨大的错误。在备份与恢复时，切记“在学会跑之前必须先学会走”。最后将介绍您需要了解的有关 RMAN 和数据库恢复的基础知识。如果您想学习这方面的更多知识，可以参阅 Oracle 出版社出版的相关书籍。可以在 www.Oraclepressbooks.com 中找到这些书籍。

1.1 备份和恢复概要

首先概括介绍备份和恢复。在精心制订计划来应对数据库可能出现的故障时，通常需要处理两个不同的问题：一个有关体系结构方面的问题是高可用性，该问题与随之而来的另一个问题备份和恢复具有松耦合关系。现在将详细讨论高可用性与备份和恢复

问题。

1.1.1　高可用性

高可用性(High Availability，HA)指的是一种防止用户发现部分或整个系统(如数据库、网络和硬件等)出现故障的体系结构。HA 解决方案包括的要素可以是镜像驱动器、RAID 体系结构、数据库群集、数据库故障切换(failover)模式，当然还包括备份和恢复。HA 增加了整个数据库体系结构解决方案的开销，且超过了所选的备份和恢复解决方案的花费。虽然 RMAN 不是一个 HA 解决方案，但它是整个数据库解决方案的一部分，而该解决方案可以包括 HA。HA 解决方案不能取代数据库备份与恢复，但是如何对数据库进行备份与恢复的策略构成了您制定高可用决策的基础。必须明确的是，如何备份和恢复数据库是 HA 决策的一部分。

如果您对 HA 解决方案有兴趣，那么可以参阅下面的几个 HA 解决方案：

- Guard/Stand-by Database
- Real Application Clusters
- Oracle Replication/Streams
- RAID 和镜像驱动器

其他供应商也提供了不同的 HA 解决方案。由于 HA 选项是一个独立于 RMAN 的主题，因此在本书中不讨论此主题。介绍 HA 解决方案的书籍有：清华大学出版社引进并出版的《Oracle Database 10g 高可用性实现方案——运用 RAC、Flashback 和 Data Guard 技术》，该书也是由本书作者之一 Matthew Hart 先生撰著的。

1.1.2　备份和恢复

接下来将介绍备份和恢复，这将进入本书的主题 RMAN。贯穿本章将详细介绍 Oracle 中的各种备份，不过现在将介绍主要的备份类型：脱机备份(冷备份)和联机备份(热备份)。

脱机备份在数据库关闭时工作，此时用户不能使用数据库。而联机备份则在数据库运行时工作，此时用户仍然可以继续使用数据库来处理事务。RMAN 支持这两种备份类型。事实上，正如您将在后面的章节中所看到的一样，RMAN 的某些功能使其成为执行联机数据库备份的较好选择。

另外，备份数据库的时机不应随意选取。对于产品数据库来说尤其如此，因为用户对数据保护有较高的期望。在决定何时以何种方式备份数据库之前，应当收集用户的需求，并考虑公司的一般备份策略。只有在收集这些需求之后，才可以实施备份计划。下面将进一步讨论如何收集这些需求。

1. 收集备份与恢复策略需求

在收集用户需求时，需要从这些需求中发现用户需要什么。并且需要向用户提出许多问题，作为一名数据库管理员(Database Administrator，DBA)，应当由您来提出这些问题。为了收集备份和恢复需求，DBA 要向用户提出下面的一系列问题：

- 在数据库发生故障时，丢失多少数据是可承受的？

- 恢复数据库时，能够忍受的最长时间是多少？
- 为了确保您的数据能够被恢复，愿意投入多少成本？
- 备份期间可以关闭系统吗？
- 受损硬件设备的替换需要多长时间？

现在，我们将详细讨论每一个问题。

可以承受的数据丢失量

这可能是上述问题中最重要的问题。所有的备份和恢复计划都有丢失数据的风险，当想采用零数据丢失解决方案时，备份和恢复计划的成本会非常昂贵。就像选用 HA 解决方案一样，机构需要量化数据丢失的成本，并且在此基础上实施成本效益最高的备份和恢复计划。客户必须知道所选择的备份和恢复计划会带来怎样的数据丢失风险。当然，每个数据库都有一个可接受的丢失量范围，有时，一个数据库所允许的数据丢失量可能比其他数据库大很多。

恢复数据库时允许的最长时间

不同的技术以不同的方式执行，所需的成本也是不同的。通常情况下，要求恢复速度越快，技术所需的成本就越高。例如，直接从磁盘上恢复数据库的成本比从磁带上恢复数据库的成本要高，但是它的恢复速度更快。对于客户来说，了解出现完全停机时恢复数据库所花费的时间是非常重要的。针对某些不能恢复数据库的例子，则可以考虑 Oracle stand-by Database 这样的技术。

恢复所花费的成本

要知道，在可接受的数据丢失量、恢复数据库所用的时间和提供给定的保护等级所花费的成本之间存在直接的关系。在开始阶段，了解客户愿意在体系结构上投入多少成本来支持备份和恢复计划，这是相当重要的。如果您建议了一个高成本的复杂体系结构，而客户只是看一眼预计的成本然后付之一笑，那将是多么让人难堪的事情。

备份期间是否关闭系统

另一个要了解的主要信息是备份期间数据库所需的状态。备份时是否能够接受停机，或者备份时需要联机吗？这个问题的答案会影响总体成本和备份策略的选择。

替换受损硬件设备需要多长时间

这是一个值得考虑的重要问题。通常不是由于数据库，而是由于硬件引起的故障。硬件故障将会严重影响数据库重新运行的时间。DBA 应当使系统各利益相关方理解硬件故障的影响并考虑使用有助于防止产生硬件故障的结构，例如采用 Oracle Real Application Clusters。

2. 精心设计备份与恢复计划

收集用户需求之后，就可以开始设计备份与恢复计划。需要解决下面这些问题：

- 基于用户(和业务)需求，需要完成数据库的脱机备份还是联机备份？
- 如果使用联机备份，那么需要多久备份一次归档的重做日志？如何在备份会话期间确保归档的重做日志不丢失？
- 公司关于可恢复性的策略和标准是什么？
- 在灾难事件中如何确保系统是可恢复的？
- 是否需要制定体系结构方面的决策？

上面的每个问题都非常重要。设计时要重点考虑到灾难事件，因为灾难事件确实会发生。公司策略有可能会代替用户的需求。这时，实现并执行备份策略和标准非常重要。因为管理一个数据库的备份和恢复策略往往比较容易，但管理多个具有不同备份和恢复方法的数据库则十分麻烦和危险。

管理归档的重做日志非常重要，这是由于归档的重做日志对于恢复来说至关重要，并且能够最大限度地支持用户需求。毕竟，用户才是我们存在的理由！为了能够决定如何设计备份策略，需要了解 Oracle 的工作原理以及 Oracle 备份和恢复的工作原理。稍后将简单讨论这些内容，首先介绍一些基本的 Oracle 术语。

1.2　需要掌握的 Oracle 术语

由于组件相互关联，因此在讨论 Oracle 体系结构时通常不知如何着手。比如介绍一个组件时，如果不引用其他组件就会相当困难。因此，本节先介绍了一些常用术语，并简要给出了这些术语的定义。这些术语将贯穿在本书的剩余章节中，因此必须完全理解这些术语的含义(在后面还将详细定义这些术语)。因此，如果您对 Oracle 内部术语的认识比较模糊，请复习下面这些术语，直至清楚地理解它们的含义：

- **警告日志(alert log)**：一个文本日志文件，记录数据库的错误和状态消息。警告日志是判断数据库故障种类的一个关键结构。通常情况下，警告日志位于后台转储目标目录中，该目录是由数据库参数 BACKGROUND_DUMP_DEST 定义的，其中警告日志称为 alert<sid>.log。
- **归档的重做日志(archived redo log)**：数据库在 ARCHIVELOG 模式下时，每次 Oracle 通过 LGWR 进程切换联机重做日志时都会生成归档的重做日志。数据库恢复期间也会使用归档的重做日志。归档的重做日志的副本可以写入 10 个不同的目录，这些目录是由数据库参数文件中的 Oracle 参数 LOG_ARCHIVE_DEST_n 定义的。同样，Oracle Database 11g 允许在称为闪回恢复区(flash recovery area)的新位置中存储归档的重做日志，在第 3 章中将详细讨论闪回恢复区。
- **备份控制文件(backup control file)**：控制文件的备份是由命令 alter database backup controlfile to 'file_name'或命令 alter database backup control file to trace 生成的。

- **块(block)**：Oracle 中最小的存储单元。默认的块大小由数据库参数文件中的参数 DB_BLOCK_SIZE 决定，并且创建数据库后该设置不变。Oracle Database 11g 允许表空间具有不同于默认值的块大小。
- **检查点(checkpoint)**：一种数据库事件，使数据库将脏(已使用)数据块从内存移出并写入磁盘。
- **数据库(database)**：由构成 Oracle 数据库的不同组件(表空间、重做日志等)组成。数据库不同于实例。数据库用于存储数据，使用 RMAN 可以备份和恢复数据库。
- **数据库一致性(database consistency)**：这是指数据库中的每个对象在某个相同的时间点是一致的。其含义是数据库数据文件中的数据在某个相同的时间点保持一致，也意味着数据库控制文件与数据库数据文件头指针保持同步。
- **数据库控制文件(database control file)**：数据库控制文件存储与数据库有关的元数据。包含关于数据库数据文件，归档重做日志，RMAN 备份等有关的信息，以及其他数据库内部信息。
- **数据库数据文件(database datafile)**：与表空间相关的物理实体。一个数据库至少含有一个数据库数据文件(该文件属于 SYSTEM 表空间)。大部分数据库是由多个不同的数据库数据文件组成的。尽管一个表空间可以有许多关联的数据库数据文件，但是一个给定的数据库数据文件只能有一个关联的表空间。
- **数据库参数文件(database parameter file)**：含有实例和数据库配置信息，数据库参数文件分为两部分(各部分相互独立)：一部分是文本文件 init.ora；另一部分是 spfile.ora 文件，该文件允许通过 alter system 命令来维持数据库参数的设置。
- **闪回恢复区(Flash Recovery Area，FRA)**：磁盘的可选配置区域，用于存储各种与恢复相关的文件。RMAN 备份文件、归档的重做日志、联机重做日志，以及控制文件都可以存储在这个区域中。可以在第 2 章中看到关于 FRA 的更多信息，第 3 章中则有它的相关的设置信息。在本书的大多数章节中都会看到关于 FRA 使用的示例。
- **区组(granule)**：Oracle 连续内存单位。所有的系统全局区(System Global Area，SGA)其内存分配都近似为最接近的区组单位。区组大小由预计的整个 SGA 内存决定，可能为 4MB 或 16MB。当 Oracle 使用较大的区组大小时，大于 128MB 的 SGA 就可能是断点。分配给数据库的区组数量在数据库启动时确定。
- **实例(instance)**：一个实例是 Oracle 内存和进程的集合。若已分配 SGA 内存，且相应的 Oracle 进程均启动和成功运行时，就可以说启动了 Oracle 实例。需要注意的是，Oracle 实例运行不表示数据库自身被打开。在给定时间，一个实例只能与一个数据库关联。
- **联机重做日志(online redo log)**：发生重做时，重做会被实际存储在数据库的联机重做日志中。Oracle 要求至少为一个数据库的操作创建两个联机重做日志。这些联机重做日志可以含有多个保护重做的镜像副本。这称为多路重做日志。当一个联机重做日志写满重做内容时，Oracle 会切换到下一个联机重做日志，这称为日志切换(log switch)操作。

每个联机重做日志文件都有关联的唯一日志序列号。一个日志序列号唯一地标识一个联机重做日志文件，如果这个联机重做日志文件已归档，那么就标识相关联的归档的重做日志文件。通过 V$LOG 视图查询可以查找到联机重做日志的日志序列号，在 V$ARCHIVED_LOG 视图或 V$LOG_HISTORY 视图中可以查找给定的归档重做日志的日志序列号。

此外，一个联机重做日志(以及一个归档的重做日志)包含该日志文件独特的一组数据库系统更改号(System Change Number，SCN)。在恢复期间，Oracle 按日志序列号的顺序来应用联机重做日志(归档的重做日志)中的撤消(undo)操作。

- **进程(process)**：执行 Oracle 数据库实际操作的程序。在 Oracle Database 11*g* 中有 5 种必需的进程，也有其他一些进程。
- **重做(redo)**：记录对给定数据库的所有修改。对于数据库中的几乎每一个修改来说，都会生成关联的重做记录。
- **模式(schema)**：拥有 Oracle 中的多种逻辑对象(如表和索引)，实际上与用户的意义相同。
- **系统全局区(System Global Area，SGA)**：Oracle 启动时分配的共享内存区，所有 Oracle 进程都共享这部分内存。
- **系统更改号(System Change Number，SCN)**：一个表示某时间点上数据库当前状态的计数器。与 VCR(录像机)的计数器相同，随着时间的推移，SCN 会相应地增加。每个 SCN 代表一个数据库生命周期中的一点。因此，在上午 11 点，数据库 SCN 可能是 10ffx0(十进制数 4351)，而在中午 12 点，数据库 SCN 则可能是 11f0x0(十进制数 4592)。
- **表空间(tablespace)**：一个物理逻辑实体。由于它存储 Oracle 逻辑对象(如表和索引)，所以是逻辑实体；由于它由一个或多个数据库数据文件组成，所以又是物理实体。一个数据库必须至少包含一个表空间(即 SYSTEM 表空间)，不过大部分数据库包含多个不同的表空间。
- **跟踪文件(trace file)**：数据库会在许多不同的情况下生成跟踪文件，这些情况包括进程错误。每个数据库进程也会生成自己的跟踪文件。在试图解决数据库故障问题时，跟踪文件会起到重要的作用。

1.3　数据库软件的控制

在执行各种恢复操作期间，需要对 Oracle 数据库和它的相关实例的状态进行控制。下面先简单回忆一下启动和关闭 Oracle 数据库的方法。

使用 SQL*Plus Oracle 实用程序可以启动 Oracle Database 11*g* 数据库。此时，可以使用 SYSDBA 的登录 ID 来登录用户系统，然后在 SQL*Plus 提示符下执行 startup 命令，如下所示：

```
/usr/Oracle>sqlplus / as sysdba
SQL*Plus: Release 10.2.0.1.0 - Production on Mon Jan 26 10:55:49 2010
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters,Automatic Storage
Management, OLAP,Data Mining and Real Application Testing options
Connected to an idle instance.
SQL> startup
```

用 startup 命令启动 Oracle 数据库时，执行的操作要经历 3 个不同的阶段：

- **实例启动(instance startup)**：启动 Oracle 数据库实例。
- **数据库加载(database mount)**：加载 Oracle 数据库。
- **数据库打开(database open)**：打开 Oracle 数据库以供用户活动。

注意：
RMAN客户端(后面章节中将讨论)有能力自己关闭和启动Oracle数据库。在大多数情况下，不需要在恢复操作期间从 RMAN 转移到 SQL *Plus。

startup 命令有几种不同的变体(对于了解多个不同的 RMAN 操作非常重要)，这些变体包括：

- **startup**：使 Oracle 经历 3 个启动阶段，并且打开到用户社区。
- **startup restrict**：使 Oracle 经历 3 个启动阶段，并且以限定模式打开数据库。只有具有限定权限的用户才可以访问数据库。
- **startup nomount**：在成功启动数据库实例后停止启动进程。通常在创建数据库之前使用该命令来启动数据库实例。如果需要重新创建控制文件，使用这条命令也会非常方便。需要注意的是，为了在给定数据库中使用 RMAN，必须能够使用 startup nomount 命令成功地启动实例。
- **startup mount**：在成功启动数据库实例并加载数据库后，停止启动起程。如果需要恢复 SYSTEM 表空间，这个命令非常有用。
- **startup read only**：使 Oracle 数据库(或备用数据库)以 READ ONLY 模式打开。因此，不支持 DML 操作，但可以查询数据库。如果正在执行时间点恢复操作，并且希望确保在使用 resetlogs 命令提交新数据库实例之前将数据库恢复到当前的时间点，这个命令非常有用。
- **startup force**：与 shutdown abort 命令(接下来会讨论该命令)一起关闭数据库。startup force 命令后面可以再次加上数据库的打开模式。示例如下：
  - startup force restrict
  - startup force mount
  - startup force nomount

现在已经知道启动数据库的方法，但是，怎样关闭数据库呢？同样，可以在 SQL*Plus 提示符下执行 shutdown 命令，该命令有下面几种形式。

- shutdown(还有 shutdown normal)：使 Oracle 等待直至所有用户进程与数据库断开连接。所有进程断开连接之后，数据库将会完全关闭。使用这个命令选项可以避免实例恢复。执行 shutdown 命令后，任何新的用户进程都不能连接到数据库。
- shutdown immediate：清除所有存在的用户会话，并且回滚所有未提交的事务。使用这个命令选项可以避免实例恢复。执行 shutdown immediate 命令后，任何新的用户进程都不能连接到数据库。
- shutdown abort：通常会使数据库崩溃。使用这个命令选项要求是实例恢复(而不是介质恢复)。执行 shutdown abort 命令后，任何新的用户进程都不能连接到数据库。
- shutdown transactional：使 Oracle 等待所有用户进程提交当前事务，然后断开这些用户进程并关闭数据库。在等待完成这些事务的期间，任何新的用户会话都不能连接到数据库。

由于本书频繁使用到这些命令，因此很有必要了解执行这些命令之后数据库和相关实例的状态。

1.4　Oracle 体系结构

下面开始介绍 Oracle 的物理组件。首先介绍构成 Oracle 数据库的进程，然后介绍 Oracle 内存结构以及 Oracle 数据库的不同逻辑结构、物理结构和物理逻辑结构，最后再讨论实例和 Oracle 数据库之间的差异。

1.4.1　Oracle 进程

执行 startup nomount 命令时，Oracle 会试图启动 Oracle 实例。当所需的操作系统进程(程序)被启动，且分配了一个称为系统全局区(SGA)的内存区之后，Oracle 实例就被启动。本小节将介绍启动 Oracle 的进程。首先介绍的是实现 Oracle 数据库功能所需的 5 个基本进程，然后介绍用户进程和服务器进程，最后还将介绍一些其他的 Oracle 进程。

注意：
这里只简要介绍了 Oracle 进程。如果要获取更详细的信息，可以参阅 Oracle 的相关文档。

1. 五种必需的 Oracle 进程

如果成功启动了一个 Oracle Database 11*g* 实例，就至少启动了 5 种不同的进程。当然，在特定的系统中(如基于 Microsoft 的各种操作系统)，这 5 种不同的进程只是单个 Oracle 进程中的线程，不过其基本原理是相同的。下面是这些必需的进程：

- PMON：也称为进程监控进程(process monitor process)，是一个“Jamaican 进程”。
- SMON：也称为系统监控进程(system monitor process)，是另一个“Jamaican 进程”。

- DBWn：称为数据库写入器进程(database writer process)。在 Oracle Database 11g 中，最多可以用 9 个这样的进程来配置一个实例(不过通常只需要一个这样的进程)。DBWn 负责将信息从 SGA 的数据库缓存区写到数据库数据文件中。
- LGWR：日志写入器进程(log writer process)负责将生成的重做信息从日志缓冲区写到数据库联机重做日志中。用户会话提交时会通知 LGWR 写入信息，当重做日志缓冲区快被写满时也需要通知 LGWR 进程。
- CKPT：在执行检查点操作时，CKPT 进程会向 DBWn 通知该检查点。CKPT 进程也用当前的检查点信息来更新数据库数据文件头。

2. 用户进程和服务器进程

用户连接到数据库时会生成一个用户进程(或者是一个在 Windows NT 上启动的新线程)，该进程将与另一个生成的服务器进程连接。这些进程使用多种协议(如 Bequeath 协议或 TCP/IP 协议)相互通信。

3. 其他可选的 Oracle 进程

根据 Oracle 数据库参数文件的配置，启动 Oracle 实例时可能会运行许多其他的 Oracle 进程(在某些情况下，可选的进程可能会在请求后很长一段时间后才启动)。由于大多数可选的进程与 RMAN、数据库备份与恢复都没有关系(除非该进程的故障会导致数据库崩溃，但这种情况很少发生)，因此不必过多地讨论这些进程。Oracle 的相关文档(可通过 otn.Oracle.com 联机获得)中收录了所有可选的进程，此外 Oracle Press 出版的书籍也介绍了一些有关这方面的知识。

与 RMAN、数据库备份和恢复有一定关系的一种可选进程是 ARCHn 进程。如果需要进行联机备份，这种进程(事实上可能存在一个或多个这样的进程)就是备份与恢复进程的关键因素。参阅本章后面的 1.6.1 节，可以更深入地了解 ARCHn 进程。

1.5 Oracle 内存和 RMAN

本小节将介绍与 RMAN 有关的内存区。与任何进程相同，RMAN 也需要使用内存，这些内存用于 RMAN 操作，也用于数据库交互。首先，我们将详细介绍 Oracle SGA，接着会介绍私有全局区(Private Global Area，PGA)。

Oracle 系统全局区

就 RMAN、备份和恢复而言，我们所关心的主要内存结构称为系统全局区(SGA)。SGA 是一块大的共享内存分配区，它可以分为以下几个内存子结构：

- 数据库缓存区(database buffer cache)
- 共享池(shared pool)
- 重做日志缓冲区(redo log buffer)
- 大池(large pool)

- Java 池(Java pool)
- 流池(Streams pool)

其中，RMAN 用户最关心的是共享池和大池。RMAN 使用一些 Oracle PL/SQL 程序包来执行操作(第 2 章将介绍这些内容)，这些程序包与其他 Oracle PL/SQL 程序包相同，必须加载到共享池。如果共享池不够大或者共享池结构零散，就有可能不能够执行 RMAN 程序包。因此，为共享池分配用于 RMAN 操作的足够内存至关重要。

RMAN 在特定情况下才使用大池，即使配置了大池，RMAN 也不会在默认情况下使用大池。如果 BACKUP_TAPE_IO_SLAVES 或 DBWR_IO_SLAVES 数据库参数中的任一参数被设置为 TRUE，那么 RMAN 就允许双路 RMAN 备份(或者在不同位置上并行生成同一个备份的副本)。如果希望模拟异步 IO，这些参数通常设置为 TRUE。这时，Oracle 使用的是大池内存，而不是本地内存。默认情况下，Oracle 使用的是进程全局区(Process Global Area，PGA)，因此没有必要为很少使用的大池配置大量的内存。

在 SGA 中定义内存分配

基于数据库参数文件中的参数设置，Oracle 分配了各 SGA 组件的大小。这些参数包括 MEMORr_MAX_SIZE、MEMORr_TARGET、SGA_MAX_SIZE、SGA_TARGET、SHARED_POOL_SIZE、DB_CACHE_SIZE、DB_nK_CACHE_SIZE、LOG_BUFFER、LARGE_POOL_SIZE 和 JAVA_POOL_SIZE(以及其他一些参数)，基于使用的不同的版本可能会有些差别。Oracle 文档中有这些参数的定义，可以参阅这些文档来了解相关内容。与 RMAN 使用有关的主要参数有 SHARED_POOL_SIZE 和 LARGE_POOL_SIZE，第 2 章将详细介绍这两个参数。

前面简要讨论了 Oracle 实例的组成部分，介绍了不同的 Oracle 进程和 Oracle 内存结构，这些进程和内存组合在一起就形成了一个 Oracle 实例。现在我们创建了一个 Oracle 实例，可以准备讲述数据库了。下一节将讨论组成 Oracle 数据库的多种结构。

1.5.1　Oracle 数据库

下面将介绍 Oracle 数据库自身的体系结构。一个 Oracle 数据库由许多不同的结构(包括一些逻辑结构、物理结构和物理逻辑结构)组成。本小节将介绍每种结构类型，并讨论 Oracle 数据库的每个组件。最后还将介绍闪回恢复区(Flash Recovery Area，FRA)和自动存储管理(Automatic Storage Management，ASM)来结束本节。

Oracle 物理组件

Oracle 数据库物理体系结构包括以下组件：

- 数据库数据文件
- 联机重做日志
- 归档重做日志
- 数据库控制文件
- Oracle 表空间

- 闪回日志(可选组件)

上面的每种组件都被物理定位在与计算机连接的存储设备上。这些对象构成了 Oracle 数据库的物理实体，为了恢复数据库还可能需要从备份中还原和恢复一个或多个这样的对象(除了闪回日志)。下面将详细介绍这些对象。

数据库数据文件

数据库数据文件是数据库的数据存储介质，这些数据文件与表空间相联系，后面将会看到这一点。在数据库中存储信息时，信息最终会存储在这些物理文件中。每个数据库数据文件都包含一个数据文件头，数据文件头中的信息帮助跟踪这个数据文件的当前状态。数据文件头在检查点操作期间被更新，并反映这个数据文件的当前状态。

数据库数据文件可以具有许多指定的不同状态。我们最关心的主要状态是 ONLINE 和 OFFLINE 状态，ONLINE 是正常状态，OFFLINE 则是非正常状态。数据库数据文件还可以是 RECOVER 状态，该状态表示这个数据文件存在问题并且需要被恢复。

如果数据库处于 ARCHIVELOG 模式下(后面将详细介绍该模式)，可以使数据文件脱机，这可能是特定的恢复操作所需要的。如果数据库处于 NOARCHIVELOG 模式下，只能删除数据库数据文件才能使这个数据文件脱机。因为脱机删除数据文件会给数据库造成坏影响(如数据丢失)，所以需要谨慎处理。

联机重做日志

如果将 Oracle SCN 比作 VCR 上的计数器，那么重做日志就好比录像带。联机重做日志负责记录数据库中发生的每个原子操作更改。考虑到性能和数据保存因素，每个 Oracle 数据库都必须至少有两个不同的联机重做日志组，大部分数据库通常都具备两个以上的联机重做日志组。

出于保护的目的，每个联机重做日志组都可以拥有多个位于不同磁盘驱动器上的日志。Oracle 将并行地写入不同的日志，从而使得写入进程更有效。Oracle 以轮询方式每次写入一个重做日志组。当一个重做日志组被写满时，LGWR 进程会关闭这些重做日志并打开下一个联机重做日志进行处理。

重做日志中的记录称为更改矢量(change vector)。每个更改矢量以 SCN 顺序来代表一个原子数据库更改，这样可以在恢复(RMAN 恢复或手动恢复)期间将所有更改记录按顺序应用于数据库，从而实现恢复故障时间点(或者要求的其他较早时间)的数据库。LGWR 进程负责将更改矢量(累积在一起就是重做)从重做日志缓冲区中写入到联机重做日志中，本章的 1.7 节将详细讨论这个问题。

归档的重做日志

Oracle 结束一个联机重做日志的写入并开始写入另一个联机重做日志时，会发生日志切换(log switch)。如果数据库处于 ARCHIVELOG 模式下并且正在运行 ARCH 进程，日志切换的结果是生成联机重做日志的一个副本。这个联机重做日志的副本称为归档的重做日志。Oracle 可以将归档的重做日志文件最多复制到 10 个不同的目录。在介质恢复

期间，数据库使用归档的重做日志来执行恢复操作，在 1.7 节中将详细讨论这个问题。

数据库控制文件

每个 Oracle 数据库都有一个或多个数据库控制文件。控制文件中包含了许多数据库信息，如当前 SCN、数据库数据文件的状态以及数据库状态。另外，控制文件也存储多种 RMAN 操作的关键信息，如每个数据库数据文件的备份状态。如果丢失控制文件，则需要遵循特定的过程来在控制文件中重新创建 RMAN 类别。在控制文件中还包含一类 DBA 感兴趣的关于检查点 SCN(有关最近修改的给定数据文件的 SCN)的信息。Oracle 将对该检查点 SCN 与存储在数据文件头文件中的检查点 SCNs 进行交叉检查。如果它们完全匹配，则不需要进行数据库恢复工作。反之则需要进行某种类型的恢复工作，例如当系统崩溃时进行的自动恢复。有时需要进行介质恢复，例如当数据文件丢失时。

Oracle 表空间

Oracle 表空间是 Oracle 的一种非实体部分。表空间是 Oracle 物理世界(表现形式为数据库数据文件)和逻辑世界(表现形式为 Oracle 表空间)之间的链接。我们通常将表空间当作物理-逻辑结构。Oracle 在表空间中存储对象，如表和索引。

表空间由一个或多个 Oracle 数据库数据文件组成，因此一个表空间可用的总空间取决于所有数据库数据文件分配的总空间的大小。表空间可以是 OFFLINE 状态或 ONLINE 状态，还可以处于 READ WRITE 或 READ ONLY 模式下。如果表空间处于 READ ONLY 模式下，表空间的内容就不会更改。因为 READ ONLY 表空间的内容不会更改，所以 DBA 通常在表空间被设为只读后只备份一次 READ ONLY 表空间的数据库数据文件。当然，如果这个表空间不再处于 READ ONLY 模式下，就需要再次启动表空间备份。

闪回日志

Oracle Database 10*g* 引入了将 Oracle 数据库闪回到其他时间点的功能。通常要用闪回日志来帮助实现该功能。闪回日志存储在 FRA 中。Oracle 单独负责管理闪回日志，因此它将根据需要创建、删除和调整闪回日志的大小。同样要注意，Oracle 没有归档闪回日志，因此不需要进行恢复。RMAN 支持闪回恢复。

闪回恢复区

Oracle Database 10*g* 引入了 FRA 的概念，它允许用户在磁盘上定义一块中心区域以用来恢复相关文件(例如 RMAN 备份和归档的重做日志)。尽管 FRA 的确参与闪回数据库(Flashback Database)操作，但请注意不要将闪回恢复区与 Oracle 的闪回数据库特性混淆，FRA 比闪回数据库的功能要强大得多。下面的结构可存储在 FRA 中：

- 归档的重做日志
- RMAN 备份固定形式
- RMAN 数据文件副本
- 闪回日志

- 数据库控制文件的副本
- 每个重做日志组的一个成员
- 控制文件自动备份和副本

我们将在第 2 章和第 3 章中更详细地介绍 FRA。

Oracle 自动存储管理(ASM)

Oracle ASM(Automatic Storage Management)是 Oracle 用于管理数据库文件的集成系统。ASM 支持多种不同的文件系统类型，包括已使用的磁盘驱动器、原始磁盘驱动器和 NetFiler 设备等。ASM 的思路是通过让 Oracle 负责基本的磁盘管理操作(例如负载平衡和数据保护)来简化 DBA 的工作。RMAN 支持 ASM 基础架构，通过该基础架构可将数据库 FRA 放置在 ASM 磁盘上，或者可以直接备份到 ASM 磁盘。

虽然 ASM 有一定的作用，但我们认为在大多数 Oracle 安装中不需要安装 ASM。如果有一个单一的、非 RAC 的、带有两个或 3 个数据库的服务器，则不需要 ASM。在本书中，我们给出 ASM 的介绍是为了让用户在使用 RMAN 和 ASM 时，可以使用 ASM 进行备份和恢复。

1.6 更多关于 Oracle 重做日志的内容

尽管已经介绍了一些 Oracle 重做日志的内容，但仍然需要进行深入的讨论。因为它非常重要，尤其是本书重点讨论的是 RMAN。您可能会产生疑问“难道 RMAN 不能为我解决所有问题吗？”RMAN 的确试图为您解决所有问题，但是您将会发现正是您的洞察力帮助您节省时间，而这种洞察力来源于您所拥有的知识。按图索骥并非最佳方式，应该理解 Oracle 备份与恢复的原理和机制。正如 Carl Jung 所说“一个人穿着合适的鞋子会夹坏其他人的脚趾。生活中不存在万能的方法。”仅仅知道备份和恢复的简单原理，您将很难面对将会出现的潜在的问题。

通常当数据库建立时创建重做日志。当数据库发生变化时，需要对联机重做日志文件进行修改，例如创建更多的重做日志，修改重做日志的大小，甚至可能会对重做日志重新命名。作为一名称职的 DBA，应该理解联机重做日志和归档重做日志。本节将详细讨论重做日志，首先对重做日志进行介绍，然后讨论多路联机重做日志组以及重做日志序列号，最后讨论联机重做日志的管理问题。

重做日志概述

Oracle 重做日志包含两类：

- 联机重做日志
- 归档重做日志

在还原和恢复 Oracle 数据库时，重做日志是最重要的组件之一。因为重做日志存储了数据库发生的所有变化。数据库操作期间，Oracle LGWR 进程负责将数据操作情况写入联机重做日志，创建您永远也不希望使用的更改记录。

LGWR 进程负责将由 Oracle 事务建立的称为 redo 的信息写入联机重做日志文化。Redo 是发生在数据库中的事件以及事件发生顺序的记录。几乎所有的 DML, DDL, 事务提交操作都会创建 redo。在数据库恢复期间，Oracle 读取 redo 并按照 redo 建立的顺序恢复数据库。一般该恢复过程在后台进行，不需要 DBA 参与(例如崩溃恢复)，但有时情况有所不同，例如在数据库恢复和数据文件恢复时，需要 DBA 参与其中。

联机重做日志文件大小固定。当 LGWR 进程达到给定的联机重做日志文件末端时，它将会关闭该文件，并寻找另外的联机重做日志文件并写入。该进程称为日志切换(*Log Switch*)。日志切换是一个串行进程，从性能角度来看，该进程开销昂贵。由于本书并不讨论性能问题，所以不对日志切换的性能问题开展更加深入的讨论。

在日志切换期间，LGWR 寻找可写入的联机重做日志文件。如果找到可用的联机重做日志文件，LGWR 将打开该文件并开始写入 redo。如果 LGWR 不能找到可用文件，则等待联机重做日志文件，直到有可用的为止。在等待期间，LGWR 将不厌其烦地向报警日志和其他地方发送“抱怨信息”，此时对数据库的操作将被挂起，也就是说数据库将暂时停止工作，通常应该尽可能地避免这种情况发生。

每个创建的联机重做日志文件分配给一个联机重做日志组。在非集群配置环境下，Oracle 在同一时间仅向一个重做日志组写入日志。如果运行在 RAC(Real Application Clusters)环境中，每个 RAC 实例将 redo 写入它自己的重做日志组中。

联机重做日志组具有几个不同的状态：

- Current 正在使用的联机重做日志。
- Active 处于该状态的联机重做日志不在当前重做日志文件组中，它正在等待归档进程(ARCH Process)将重做复制到归档重做日志中。
- Inactive 已经被归档的不活跃的联机重做日志。
- Unused 正在被 Oracle 数据库使用的联机重做日志。

通过查询 V$LOG 视图可以获得联机重做日志组的状态。

```
SQL> select group#, status from v$Log;
    GROUP#  STATUS
----------  --------------------------------
         1  INACTIVE
         2  INACTIVE
         3  INACTIVE
         4  CURRENT
```

多路联机重做日志

如果 active 联机重做日志丢失，将会造成巨大的损失。一旦发生该问题，数据库将会崩溃，也将可能面临数据的丢失，需要大量的时间进行恢复工作。因为 Oracle 数据库恢复到失败时间点的工作主要依靠联机重做日志的可用性。正如您所知，联机重做日志将使数据库易于产生磁盘设备的丢失，发出不正确的删除命令，或其他的错误。为解决这一问题，可以为每个联机重做日志创建镜像。若为一个联机重做日志建立一个以上的副本，该日志所在的组称为多路联机重做日志组。为给联机重做日志组提供多重保护，

通常将多路日志副本放在不同的物理设备中。从高可用性角度考虑，建议将联机重做日志组中的成员分别存放到不同的磁盘驱动设备上。以下是一个建立多路联机重做日志组的示例：

```
alter database add logfile group 4
    ('C:\ORACLE\ORADATA\BETA1\RED004a.LOG',
       'C:\ORACLE\ORADATA\BETA1\RED004b.LOG') size 100m reuse;
```

由于多路联机重做日志组的多个成员可以并行写入，因此在每个组中尽管由多个日志成员组成也不会带来性能问题。

日志序列号

当写入每个联机重做日志组时，该组将被分配一个序号。该序号称为日志序列号。新建立的数据库其第 1 个日志序列号总是以 1 开始。写入联机重做日志组时，该序号在发生日志切换操作后将按顺序递增。如上所述，下一个写入的联机重做日志序列号为 2，后面的以此类推。

执行正常的数据库操作时，Oracle 将打开一个可用的联机重做日志，将 redo 写入其中，一旦填满联机重做日志，则关闭它。一旦该联机重做日志被填满，LGWR 进程将切换到另一个联机重做日志组。此时，如果数据库处于 ARCHIVELOG 模式，LGWR 将激活 ARCH 进程，ARCH 进程将以顺序循环方式写入联机重做日志。如图 1-1 所示。

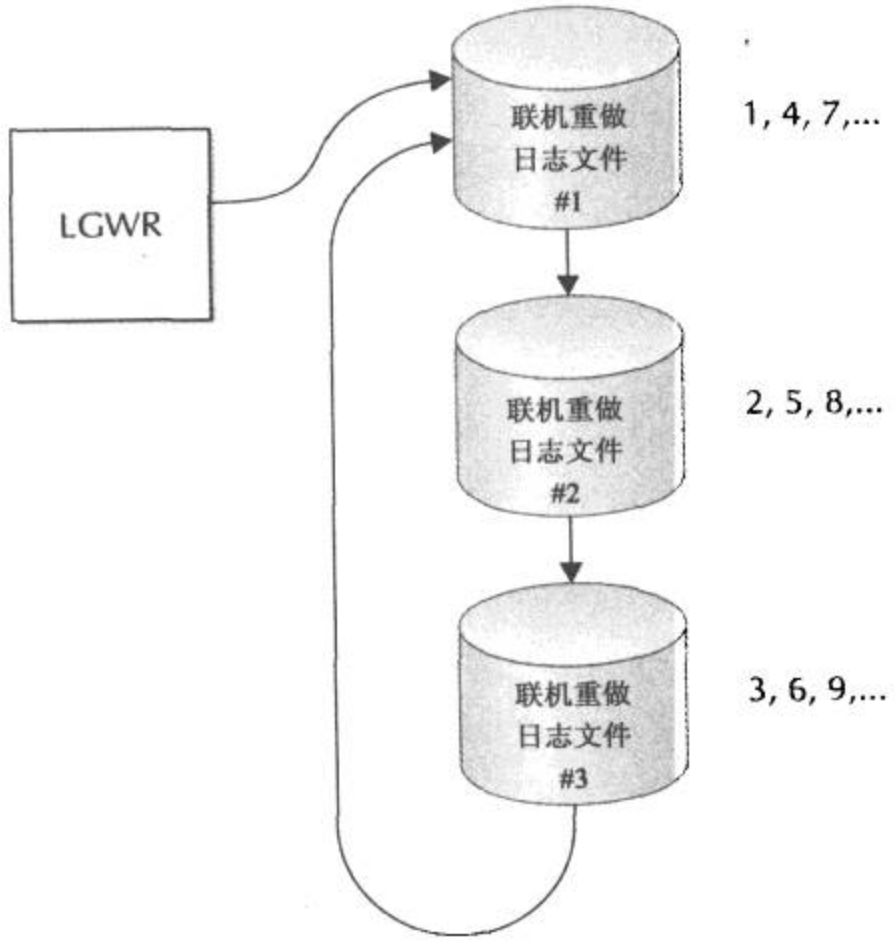


图 1-1 写入联机重做日志

ARCH 进程响应 LGWR 的调用，将在由 Oracle 数据库参数 LOG_ARCHIVE_DEST_n 定义的地址和/或者定义的闪回恢复区上创建联机重做日志的副本。直到 ARCH 进程至少创建了一个归档重做日志，Oracle 才能重用有关的联机重做日志文件。可能需要创建多个归档重做日志后，相关联的联机重做日志才能够被 Oracle 重用，这将由系统的配置所决定。在创建归档重做日志时，它们将保留分配给父联机重做日志的日志序列号。该序列号在数据库中将保持唯一性，直到执行 resetlogs 操作打开数据库为止。一旦执行 resetlogs 操作，日志序列号将重置为 1。

在执行恢复操作期间使用 resetlogs 命令启动数据库时还需要注意一点。如果使用的是 Oracle Database 10g 或者更高的版本，Oracle 提供使用执行 resetlogs 命令时间点前的备份来恢复数据库的功能。当执行 resetlogs 命令时，Oracle 将会在联机重做日志重置时归档所有未归档的联机重做日志。该项功能提供使用执行 resetlogs 命令时间点前的备份恢复数据库的功能。使用备份文件和所有的归档重做日志，可以实现跨越 resetlogs 命令的数据库恢复，该功能使 DBA 在使用基于 resetlogs 的命令恢复数据库后不用忙于立即进行备份(尽管该备份仍然是非常重要的)。该功能减少了平均恢复时间，用户可以在恢复后打开数据库，而不需要首先去备份数据库。

联机重做日志管理

alter database 命令可用于增加或删除联机重做日志。下例所示的是为数据库增加一个联机重做日志组。新的日志文件组的组号为 4，定义其大小为 100m。

```
alter database add logfile group 4
'C:\ORACLE\ORADATA\BETA1\REDO04.LOG' size 100m;
```

可以通过视图 V$LOG 和视图 V$LOGFILE 查看日志文件组的结果。

```
SQL> select group#, sequence#, bytes, members from v$log
  2  where group#=4;
    GROUP#      SEQUENCE#    BYTES    MEMBERS
----------------------------------------------------------
         4               0    104, 857, 600         1
SQL> select group#, member from v$logfile
  2  where group#=4;
    GROUP#      MEMBERS
----------------------------------------------------------
         4     C:\ORACLE\ORADATA\BETA1\REDO04.LOG
```

下面的例子中，将重做日志文件组 4 从数据库中删除。注意这并不能完全删除其物理文件。在删除日志文件组后，还需要执行该功能这将很危险，所以操作时要谨慎：

```
alter database drop logfile group 4;
```

注意：

若使用 FRA 或者设置了参数 DB_CREATE_ONLINE_LOG_DEST_n，那么在您发出删除命令后，Oracle 将删除联机重做日志。

当希望调整日志文件组大小时，需要先删除该日志文件组，然后以希望的大小重新建立该日志文件组。

1.6.1 ARCHIVELOG 模式与 NOARCHIVELOG 模式

Oracle 数据库可以在两种模式下运行。默认情况下，数据库在 NOARCHIVELOG 模式下创建，这种模式允许正常的数据库操作，但是不提供时间点恢复操作或联机备份的功能。如果要实现联机备份(或热备份)，就需要以 ARCHIVELOG 模式运行数据库。在 ARCHIVELOG 模式下，数据库通过 ARCH 进程在一个或多个归档日志目标目录中生成所有联机重做日志的副本。

使用 ARCHIVELOG 模式不仅仅是简单地将数据库置入 ARCHIVELOG 模式，还需要对数据库进行一些配置。用户还必须配置 ARCH 进程，并且提供归档的重做日志目标目录。需要注意的是，一旦 Oracle 数据库处于 ARCHIVELOG 模式下，数据库活动将在所有可用的联机重做日志被使用过之后被挂起。数据库在归档这些联机重做日志前保持挂起状态。因此，对于处在 ARCHIVELOG 模式下的数据库，错误配置将导致它因不能归档当前联机重做日志而挂起所有的操作。这听起来很严重，但实际上确实如此，归结于以下的一些基本方面：

- 适当地配置数据库(本书涉及数据库的备份与恢复的配置)。
- 确保有足够的可用空间。
- 确保操作与预期一致。例如，如果在 ARCHIVELOG 模式下的数据库中定义闪回恢复区，则要确保归档重做日志已被成功写入到该目录中。

第 3 章将深入介绍 ARCHIVELOG 模式的含义，实现(以及禁用)它的方法，以及 ARCHIVELOG 模式操作的配置。

1.6.2 Oracle 逻辑结构

Oracle 中存在一些不同的逻辑结构，这些结构包括表、索引、视图、集群、用户自定义对象和数据库中的其他对象。模式(schema)由这些对象组成，如果要存储对象，就需要从表空间中分配存储空间。

Oracle 备份和恢复策略的最终目标是能使这些逻辑结构恢复到给定的时间点。当然，使用这些逻辑结构恢复数据，使数据状态与给定时间点一致也是非常重要的。例如，如果要将一个表恢复到上午 10 点所示的状态，但是却只恢复了与该表关联的上午 9 点所对应的一个索引，这个不一致的恢复会导致可怕的影响。这个一致性恢复概念就是 Oracle 的备份和恢复机制的原则，RMAN 则完全体现了这个备份和恢复体系结构的架构。

1.7 组合情况

现在已介绍了 Oracle 数据库的多种组件(还有许多其他的组件)，下面综合地说明它们如何一起运作。首先将介绍整个数据库启动进程，随后说明数据库的基本操作。

1.7.1　数据库的启动和关闭

Eliza 是一名 DBA，她刚结束对数据库的操作，此时需要重新启动这个数据库。Eliza 启动 SQL*Plus，使用系统数据库管理员的账号作为 SYS 连接数据库，并在 SQL 提示符下执行 startup 命令，打开数据库。下面的示例说明了执行 startup 命令的结果：

```
SQL> startup
ORACLE instance started.
Total System Global Area   84700976 bytes
Fixed Size                   282416 bytes
Variable Size              71303168 bytes
Database Buffers           12582912 bytes
Redo Buffers                 532480 bytes
Database mounted.
Database opened.
```

前面介绍过执行 startup 命令后会经历 3 个不同阶段：实例启动、数据库加载以及数据库打开。现在将详细介绍这 3 个阶段。

1. 实例启动(未加载启动)

启动数据库时首先经历的阶段是实例启动。在这个阶段，Oracle 分析数据库参数文件，并且通过设法获得实例锁来确定实例还未运行。随后，Oracle 启动多种不同的数据库进程(参阅本章前面的 1.4.1 小节)，如 DBWn 和 LGWR 进程。Oracle 还将分配 SGA 所需的内存。一旦启动了实例，Oracle 会向启动实例的用户报告后台进程已经启动了实例，以及为 SGA 分配了多少内存。

如果 Eliza 执行 startup nomount 命令，那么 Oracle 会在启动实例后停止数据库启动进程。在执行某些恢复操作(如控制文件的重建)时，管理员可能会这样启动实例。

2. 数据库加载(加载启动)

启动进程中的第二个阶段是数据库加载。在加载阶段，Oracle 打开数据库控制文件。成功打开控制文件后，Oracle 从这个控制文件中析取数据库数据文件名，为打开这些数据文件做好准备。需要注意的是，这个时候 Oracle 不会检查这些数据文件是否存在，而只是从控制文件中标识数据文件的位置。完成这个步骤后，Oracle 向用户报告已经加载了数据库。

如果 Eliza 执行的是 startup mount 命令，此时 Oracle 会停止打开数据库并等待下一个指令。当 Oracle 实例已启动并且数据库已加载但未打开时，可以执行某些恢复操作，这些操作包括将数据库数据文件和恢复系统表空间数据文件的位置重命名。

3. 数据库打开

Eliza 执行的是 startup 命令，Oracle 将继续启动进程并设法打开数据库。在数据库打开阶段，Oracle 验证数据库数据文件是否存在并打开这些文件。Oracle 打开数据文件时会检查数据文件头，并且比较这些文件头中的 SCN 信息和控制文件中存储的 SCN。接下

来将讨论 SCN。

SCN 是跟踪数据库状态的 Oracle 方法。数据库发生更改时，就会生成与更改关联的 SCN。在数据库数据文件中写入更改时(发生于检查点操作期间)，数据文件头会更新当前 SCN。当前 SCN 也会被记录在数据库控制文件中。

Oracle 设法打开数据库时会检查每个数据文件和数据库控制文件中的 SCN。如果 SCN 相同并且位映射标记设置正确，就认为这个数据库是一致的，即可打开和使用数据库。

注意：

SCN 可以看作是 VCR 上的计数器。随着时间的推移，VCR 计数器会不断增加，并指向磁带所在的当前时间点。因此，如果要观看磁带上的节目，则可以将磁带倒退(或快进)到相应的计数器位置(即该节目的开始位置)。SCN 的工作原理与 VCR 计数器相同。Oracle 需要恢复数据库时会“倒退”到相应的 SCN，然后重放这个 SCN 后的所有事务，直至数据库恢复。

如果 SCN 不相同，则 Oracle 会自动执行灾难或实例恢复。如果联机重做日志文件中存在生成一致的映像所需要的重做，就会发生灾难或实例恢复。如果由于数据文件损坏或者由于恢复需要的重做不在联机重做日志中而不能执行灾难或实例恢复，Oracle 会要求 DBA 执行介质恢复(media recovery)。介质恢复需要从一个数据库备份中恢复一个或多个数据库数据文件，与实例恢复不同的是，介质恢复是一个手动执行的进程。RMAN 能够用于介质恢复，在后面的章节中将讨论这个问题。一旦数据库打开进程完全成功之后(没有恢复、灾难恢复或介质恢复)，数据库就可以用于业务。

4. 关闭数据库

当然，Eliza 在某时间点可能要关闭数据库。为了关闭数据库，Eliza 可以执行 shutdown 命令。这条命令将关闭和卸载数据库，然后几乎完全按照与前面介绍的启动进程相反的顺序来关闭实例。shutdown 命令有若干选项。

需要特别注意的是：使用 shutdown abort 命令基本上相当于模拟的数据库灾难，这条命令很常用，并且极少出现问题。但 Oracle 通常建议如果有可能，应在数据库处于一致状态的情况下关闭数据库。

如果必须使用 shutdown abort 命令来关闭数据库(在实际中，由于断电约束会经常出现这种情况)，我们就应当使用 startup 命令(或者可能是 startup restrict 命令)重新打开这个数据库。接下来，在执行任何脱机备份操作前应当使用 shutdown immediate 命令来关闭数据库。需要注意的是，由于在关闭进程期间回滚事务需要一定的时间，所以即使是用上述方法也可能会导致延迟关闭数据库。

注意：

只要备份/恢复策略是正确的，数据库在脱机备份时是处于一致状态(如正常的 shutdown 命令)还是非一致状态(如 shutdown abort 命令)并没有多大关系。Oracle 建议在数据库处于一致状态时执行冷备份，我们也建议执行这样的操作(这是因为 RMAN 不会备份联机重做日志)。最后需要注意的是联机备份根本不会使用 shutdown 命令。

1.7.2　使用数据库和内核

本小节将介绍用户在 Oracle 数据库中执行的一些不同事务。首先，将以图表的方式介绍数据库的所有进程、内存结构和数据库的其他组件；接着将描述用户对数据库的更改；然后查看提交更改及相应的操作方式。最后还要讨论数据库检查点及其工作原理。

1. 进程和数据库关系

前面已经介绍过许多不同的进程、内存结构和构成整个 Oracle 数据库的其他对象。如图 1-2 所示的图形能够帮助您更好地理解 Oracle 中不同组件之间的相互关系。

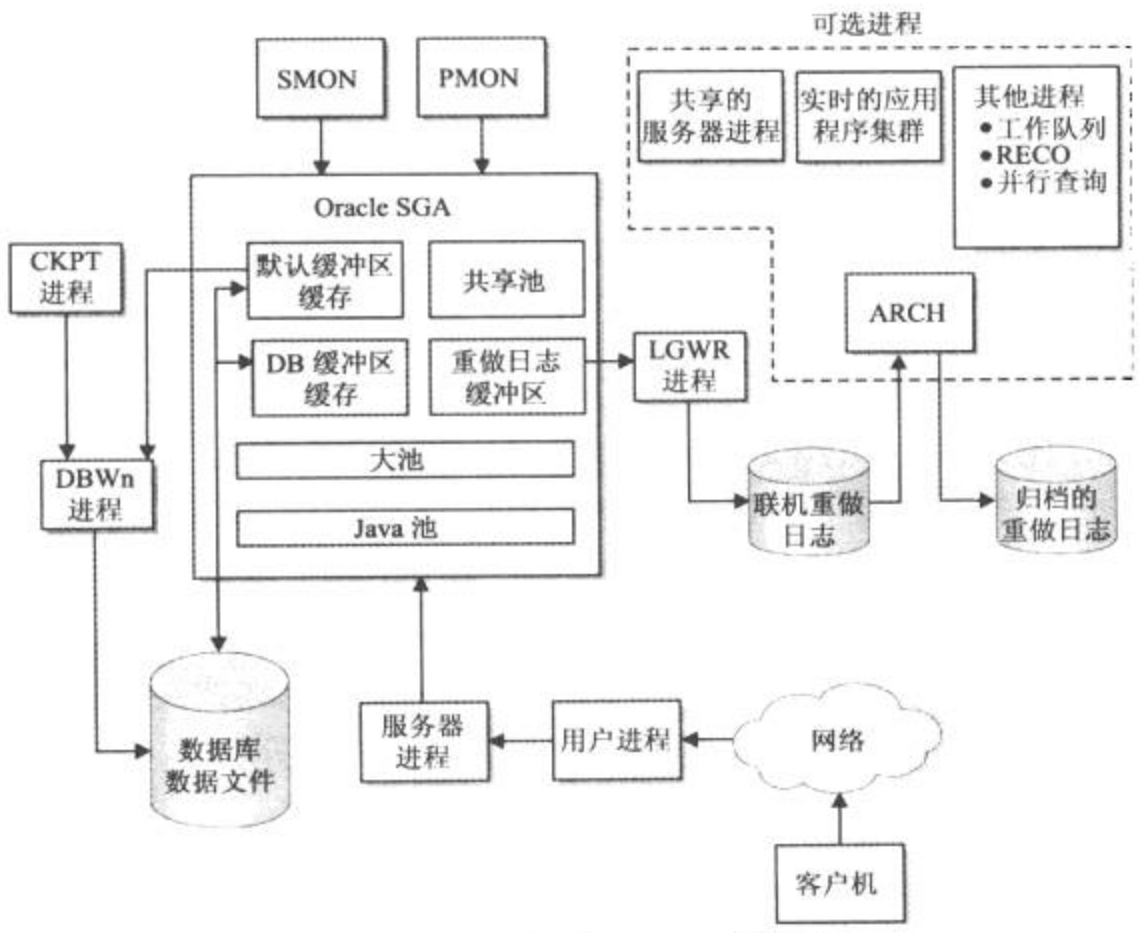


图 1-2　一个典型的 Oracle 数据库

2. 更改数据库中的数据

假设数据库已打开，用户 Fred 要在 DEPT 表中为房屋管理部门添加一条新记录。为此，他可能会执行下面的 SQL 语句。

```
INSERT INTO DEPT VALUES (60,'JANITOR', 'DALLAS');
```

insert 语句以及 update 和 delete 命令统称为数据操作语言(Data Manipulation Language，DML)。执行 insert 语句时会生成重做，并且重做被存储在 Oracle SGA 的重做日志缓冲区中。需要注意的是不管是否存在 commit 命令，此命令都生成重做。delete 和 update 命令也一样。

DML 的另一个结果是生成撤消(undo)，并且撤消被存储在回滚段(rollback segment)中。Undo 由一系列指令组成，该指令用来撤消(或回滚)已执行的语句。通过使用 Undo，Oracle 可以回滚数据库的更改，并且为其他用户提供读取一致性映像(read consistent image，也称为读取一致性)。下面将进一步介绍 commit 命令和读取一致性。

3. 提交更改

执行 insert 命令之后，为了确保这个更改被提交给数据库，Fred 还要执行 commit 命令：

```
COMMIT;
```

执行 commit 命令的作用包括：

- 对于在提交 commit 命令后的某个时间点对表进行查询的所有用户来说，这个更改是可见的。如果 Eliza 在提交 commit 命令后查询 DEPT 表，她将看到部门 60。但如果 Eliza 在提交 commit 命令前已经启动了一个查询，该查询就不会发现 DEPT 表的更改。
- 如果数据库处于 NOARCHIVELOG 模式下，那么并且要求灾难或实例恢复，这个更改就是可恢复的。
- 如果数据库处于 ARCHIVELOG 模式下(采用一种有效的备份和恢复策略)，并且要求介质恢复，同时所有归档的重做日志和联机重做日志都是可用的，那么这个更改就是可恢复的。

commit 命令使 Oracle LGWR 进程将联机重做日志缓冲区中的内容写入到联机重做日志中。其实不考虑 commit 命令，未提交的重做也会被写入到联机重做日志中(实际上未提交的更改也被写入到数据文件中)。执行一条 commit 命令时，Oracle 在重做日志缓冲区中写入一个提交矢量(commit vector)，并且在提交返回之前将缓冲区中的内容写入到磁盘上。只有在重做被成功地写入联机重做日志之后，Fred 的会话发出的提交才会返回，这个事实和上述的提交矢量确保了 Fred 所做的更改是可恢复的。

commit 命令与读取一致性

您是否注意到 Eliza 只有在 Fred 执行 commit 命令后才能查看他所做的更改？这就是所谓的读取一致性(read consistency)。读取一致性的另一个示例是 Eliza 在 Fred 提交更改之前启动报告。假定 Fred 在 Eliza 的报告期间提交了更改。在这个案例中，由于报告启动时不存在 Fred 插入的记录，所以 Eliza 的报告中如果显示部门 60 的记录，将会是不一致的。在继续运行 Eliza 的报告时，Oracle 会检查报告查询的起始 SCN，并把它和 Oracle 中生成报告输出所读取的数据块的 SCN 进行比较。如果报告的启动时间早于数据块的当

前 SCN，Oracle 会在回滚段中查找该数据块的撤消，这样可以允许 Oracle 构造一个与报告启动时间一致的映像。

Fred 在数据库上继续其他工作时，LGWR 进程按常规写入联机重做日志。一个联机重做日志在某些时间点会被写满，此时 LGWR 会关闭这个日志文件，打开并开始写入下一个日志文件。在这个过渡期间，LGWR 还会通知 ARCH 进程将刚使用的日志文件复制到归档日志备份目录中。

4. 检查点

现在，您可能想弄清更改的数据到底在何时被写入到数据库数据文件中？前面介绍过检查点是 Oracle(通过 DBWR)将数据写入到数据文件中的事件。Oracle 中存在几种不同的检查点。检查点会引发下面这些事件：

- 重做日志切换
- 正常的数据库关闭
- 表空间进入或退出联机备份模式(参阅本章后面的 1.8.2 节)

需要注意的是，由于数据库的整个生命周期内会一直标记增量检查点(incremental checkpoint)，这为数据库减少执行灾难恢复所需的总时间提供了方法。在数据库运行时，Oracle 不断地将数据流写入到数据库数据文件中，这些写操作不会阻碍数据库的运行。Oracle 提供特定的数据库参数来帮助确定 Oracle 必须处理增量检查点的频率。

1.8　Oracle 备份与恢复入门

在使用 RMAN 前，您应当理解 Oracle 中一些常用的备份与恢复概念。Oracle 中的备份通常分为两种类型：逻辑备份和物理备份。下面将简要介绍逻辑备份和恢复，然后重点介绍 Oracle 物理备份和恢复。

1.8.1　逻辑备份与恢复

Oracle Database 11g 使用 Oracle Data Pump(数据泵)体系结构来支持逻辑备份和恢复。这些实用程序包括数据泵导出程序(expdp)和数据泵导入程序(impdp)。对于逻辑备份来说，时间点恢复是不可能的。RMAN 不进行逻辑备份和恢复，因此本书不讨论逻辑备份和恢复问题。

1.8.2　Oracle 物理备份与恢复

RMAN 实际上是物理备份。本书后面的章节将深入探讨 RMAN，在这之前，我们先看看手动进行 Oracle 数据库的物理备份与恢复时的要求。RMAN 可以将我们从繁重的备份和恢复工作中解脱出来，但是它们应用的某些原理是相同的。了解手动备份和恢复的基础知识可以帮助您理解 RMAN 的工作方式，并有助于将其与以前的 Oracle 备份方法进行比较。

前面已经讨论过 Oracle 中的 ARCHIVELOG 模式和 NOARCHIVELOG 模式，在这

两种模式下，Oracle 都可以执行脱机备份。此外，如果数据库处于 ARCHIVELOG 模式，那么 Oracle 既可以执行脱机备份，也可以执行联机备份。本书后面的章节将介绍使用 RMAN 执行这些备份操作的细节。

当然，备份数据库时最好能够恢复这个数据库。在介绍有关联机备份和脱机备份的内容之后，将讨论不同的 Oracle 恢复选项。最后，在这几节中将简要介绍 Oracle 的手动备份与恢复。

1. NOARCHIVELOG 模式下的物理备份

前面已经讨论过 Oracle 数据库中的 NOARCHIVELOG 模式，这种数据库操作模式只在数据库关闭时支持数据库备份。此外，NOARCHIVELOG 模式下数据库的完全恢复只能恢复到备份的时间点。按照下面的步骤可以在 NOARCHIVELOG 模式中执行数据库的手动备份(需要注意的是，如果使用 RMAN，这些步骤会有所不同，这将在后面的章节中介绍)：

(1) 完全关闭数据库。

(2) 备份所有数据库数据文件、控制文件和联机重做日志。

(3) 重新启动数据库。

2. ARCHIVELOG 模式下的物理备份

如果在 ARCHIVELOG 模式下运行数据库，那么既可以在数据库运行时完全备份数据库，也可以在数据库关闭时进行。即使是在数据库关闭时进行备份，步骤也会稍有不同，请看如下所示的冷备份步骤：

(1) 完全关闭数据库。

(2) 备份所有数据库数据文件。

(3) 重新启动数据库。

(4) 使用 alter system switch logfile 命令强制执行一个联机重做日志切换。一旦归档了联机重做日志，那么就备份所有归档的重做日志。

(5) 使用 alter database backup control file to trace 命令和 alter database backup controlfile to 'file_name'命令创建控制文件的一个备份。

当然，在数据库处于 ARCHIVELOG 模式时，您有可能想完成数据库的联机备份(热备份)。数据库处于 ARCHIVELOG 模式时，Oracle 允许在数据库打开和运行的同时备份每个单独的表空间与其数据文件，甚至还可以在不同的时刻备份选择的数据库部分。使用下面的步骤可以执行表空间的一个联机备份：

(1) 使用 alter tablespace begin backup 命令将需要备份的表空间和数据文件置入联机备份模式。如果希望备份整个数据库，则可以使用 alter database begin backup 命令将所有数据库表空间置入热备份模式。

(2) 备份与刚被置入热备份模式的表空间相关联的数据文件(可以选择备份指定的数据文件)。

(3) 对于在步骤(1)中置入联机备份模式的每个表空间来说，都要执行 alter tablespace

end backup 命令将其从热备份模式中取出。如果希望将所有表空间从热备份模式中取出，可以使用 alter database end backup 命令。

(4) 使用 alter system switch logfile 命令强制执行一个联机重做日志切换。

(5) 一旦完成了日志切换并归档了当前的联机重做日志，就备份所有归档的重做日志。

需要注意的是，步骤(5)中的日志切换和归档的重做日志备份是必需的，这是因为恢复操作必须应用在备份期间生成的所有重做上。Oracle 在联机备份期间不断地物理更新数据文件(除了数据文件头)时，在备份操作期间存在数据块分离的可能性，这种可能性会导致备份的数据文件不一致。此外，数据库数据文件可能在备份之后、但是在整个备份进程结束之前被写入，由于备份中的每个数据文件当前可能会有不同的 SCN，因此数据文件备份映像会不一致，所以拥有在备份期间生成的重做以应用于恢复是非常重要的。

执行 alter tablespace begin backup 命令或 alter database begin backup 命令时，重做生成会发生更改。一般来说，Oracle 只将更改矢量存储为重做记录。这些小型的记录只定义已经发生的更改。当数据文件处于联机备份模式时，Oracle 会记录数据库数据文件的整个块更改，而不是仅仅记录更改矢量，这意味着联机备份期间总的重做生成会大幅增加，这样在热备份进程执行期间所需的磁盘空间和 CPU 开销会受到影响。RMAN 提供了不将表空间置入热备份模式而执行热备份的功能，这样就避免了使用额外的 I/O 操作。当结束数据文件的联机备份状态时，就可以进行正常的操作。

还需要注意的是在 ARCHIVELOG 模式的两种备份中(联机备份和脱机备份)，并不备份联机重做日志，而是备份数据库的归档的重做日志。此外，也不会备份控制文件，但是会创建备份控制文件，这是因为在恢复期间不希望冒险重写联机重做日志或控制文件。

至于为什么不恢复联机重做日志，那是因为在 ARCHIVELOG 模式的恢复期间，联机重做日志中可能有最新的重做，这样当前的联机重做日志将被用于完全的时间点恢复。由于这个原因，我们不在 ARCHIVELOG 模式的数据库恢复期间重写联机重做日志。如果数据库丢失了联机重做日志(希望不会发生这种情况)，就必须使用所有归档的重做日志来执行时间点恢复。

出于与不备份联机重做日志相同的原因，我们也不备份控制文件。因为当前控制文件包含最新的联机重做日志信息和归档的重做日志信息，所以我们不想用这些对象先前的信息来重写最新的信息。如果丢失了所有的控制文件，则可以使用备份控制文件来恢复数据库。

最后来看看对归档的重做日志文件做补充备份以及保护归档的重做日志不丢失的其他方法。丢失归档的重做日志直接影响将数据库恢复到故障点的能力。如果丢失了归档的重做日志并且联机重做日志组不再含有该日志序列号，数据库就只能恢复到丢失的归档的重做日志序列号之前的日志序列号。

3. NOARCHIVELOG 模式下的恢复

如果需要在 NOARCHIVELOG 模式下恢复一个备份，可以简单地恢复所有数据库数

据文件、控制文件和联机重做日志，然后再启动数据库。当然，完整的恢复可能还需要恢复 Oracle RDBMS 软件、参数文件和其他所需的 Oracle 项，本章的最后部分将介绍这些内容。

需要注意的是在 NOARCHIVELOG 模式下的恢复只能恢复至最后备份的时间点。如果在 NOARCHIVELOG 模式中恢复一个数据库备份，我们只能恢复该备份时间点的数据库。如果数据库处于 NOARCHIVELOG 模式，就不能恢复备份时间点后的任何更改。

4. ARCHIVELOG 模式下的恢复

使用联机备份或脱机备份都可以备份 ARCHIVELOG 模式下的数据库。与 NOARCHIVELOG 模式相比，我们可以在 ARCHIVELOG 模式下恢复故障点发生时的数据库。此外，可以选择恢复到指定时间点的数据库，还可以恢复基于更改号的指定时间点的数据库。

ARCHIVELOG 模式恢复也允许执行对数据文件、表空间或整个数据库的特定恢复。此外，还可以执行时间点恢复或指定 SCN 的恢复。我们将简要介绍这些恢复选项。

本部分将简要概述在 ARCHIVELOG 模式下数据库的完全恢复。接下来将讨论表空间和数据文件恢复，然后讨论时间点恢复。

ARCHIVELOG 模式下的完全恢复

假定数据库故障没有损害当前每个联机重做日志组中的所有成员和未备份的任何归档的重做日志，那么我们可以在 ARCHIVELOG 模式下恢复故障点的数据库备份。如果丢失了归档的重做日志或联机重做日志，就需要执行某些形式的时间点恢复，后面将讨论时间点恢复问题。此外，如果丢失了当前控制文件的所有副本，就需要恢复控制文件并执行不完全恢复。

执行下面的步骤可以在 ARCHIVELOG 模式下从数据库的一个备份中执行数据库的完全恢复：

(1) 从备份中还原所有数据库数据文件。

(2) 还原所有备份的归档的重做日志。

(3) 加载数据库(使用 startup mount 命令)。

(4) 恢复数据库(使用 recover database 命令)。

(5) Oracle 提示您应用归档的重做日志中的重做。在提示符下只需要简单地输入 AUTO，Oracle 会自动应用所有重做日志。

(6) 一旦应用了所有重做日志，就可以打开恢复的数据库(使用 alter database open 命令)。

ARCHIVELOG 模式下的表空间恢复和数据文件恢复

在数据库加载或数据库打开时可以执行表空间恢复和数据文件恢复。按照下面的步骤可以在 Oracle 数据库打开时执行一个表空间恢复：

(1) 使表空间脱机(使用 alter tablespace offline 命令)。

(2) 还原与要恢复的表空间相关联的所有数据文件。

(3) 联机恢复表空间(使用 recover tablespace 命令)。

(4) 一旦完成了恢复，使表空间联机(使用 alter tablespace online 命令)。

正如可以恢复表空间一样，还可以恢复指定的数据文件。这样做的优点是可以保持表空间联机。在恢复进程期间仅仅是驻留在脱机数据文件中的数据不可用，数据库的其他部分在恢复期间都是可用的。数据文件恢复的基本步骤为：

(1) 使数据文件脱机(使用 alter database datafile 'file_name' offline 命令)。

(2) 还原所有要恢复的数据文件。

(3) 联机恢复表空间(使用 recover datafile 命令)。

(4) 一旦完成了恢复，就使数据文件联机(使用 alter database datafile 'file_name' online 命令)。

ARCHIVELOG 模式下的时间点恢复

ARCHIVELOG 模式的另一个优点是可以将数据库恢复到指定时间点，而不是恢复到发生故障时的状态。在创建克隆数据库时(可能出于测试和报告的原因而创建克隆数据库)，或者在主要的应用程序错误和用户错误情况中，经常会用到上述功能。我们既可以将数据库恢复到指定时间点，也可以恢复到指定数据库 SCN 的状态。

如果要恢复某个时间点的表空间，则需要恢复相同时间点的整个数据库(除非执行表空间的时间点恢复，这是另一种恢复方式)。例如，假设您有一个账户管理数据库，大部分数据位于 ACCT 表空间中，并且您打算恢复两天前的数据库。因为其余的表空间(例如 SYSTEM、TEMP 和 RBS 表空间)仍然与当前时间点一致，所以您不能只是还原 ACCT 表空间并将其恢复到两天前的时间点，否则由于数据库的不一致，数据库将打开失败。

执行下面的步骤可以恢复某一时间点的数据库：

(1) 从一个备份中恢复所有数据库数据文件，这个备份在要恢复数据库的时间点之前结束。

(2) 使用 recover database until time '01-01-2010 21:00:00'命令并且应用所需的重做日志，恢复指定时间点的数据库。

(3) 一旦完成恢复，就使用 alter database open resetlogs 命令打开数据库。

也可以执行下面的步骤，选择使用 SCN 号恢复数据库：

(1) 从一个备份中恢复所有数据库数据文件，这个备份在要恢复数据库的时间点之前结束。

(2) 使用 recover database until change '221122'命令并且应用所需的重做日志，恢复指定系统更改号(SCN)的数据库。

(3) 一旦完成恢复，则打开数据库。

进一步来说，可以应用数据库的更改并在应用指定的归档的重做日志之后手动取消进程：

(1) 从一个备份中恢复所有数据库数据文件，这个备份在要恢复数据库的时间点之前结束。

(2) 使用 recover database until cancel 命令并且应用所需的重做日志，恢复指定时间点的数据库。当应用了最后一个归档的重做日志后，可以简单地执行 cancel 命令来结束重做日志应用。

(3) 一旦完成恢复，则打开数据库。

执行时间点恢复(或者有关这一问题的任何恢复)时一定要牢记数据库的一致性概念。如果要恢复给定时间点的数据库，就必须使用在这个时间点之前结束的备份来恢复数据库，还必须利用所有归档的重做日志(可能还需要剩余的联机重做日志)来完成恢复。

闪回数据库恢复

另一种可用的恢复方法是使用 Oracle 的闪回特性，这在 13 章中有详细介绍。

由于各种闪回功能使得数据库从用户级或应用级错误中恢复的时间大大减少。因此，RMAN 提供了对 Oracle Database 11g 闪回特性的支持，本书将在后面进行全面的介绍。

1.8.3 备份其他 Oracle 组件

我们已经简要介绍了 Oracle 备份和恢复的概要，最后一个问题是讨论需要备份的项。因为需要备份的项极少更改，所以这些项通常不会被频繁地备份。需要备份的项包括：

- Oracle RDBMS 软件(Oracle 主目录和 Oracle 目录清单)。
- 网络参数文件(names.ora 文件、sqlnet.ora 文件和 tnsnames.ora 文件)。
- 数据库参数文件(init.ora 文件、INI 文件等)。需要注意的是 RMAN 具有支持同时备份数据库参数文件(只有是数据库参数文件的情况下)和控制文件的新功能。
- 系统 oratab 文件和其他与 Oracle 相关的系统文件(例如用于 Oracle 的所有 rc 启动脚本)。

作为备份和恢复进程的一部分，定期地备份这些项相当重要。因为 RMAN 也不会备份这些项，所以不管执行手动备份还是 RMAN 备份，我们都需要有计划地备份上述的这些项。

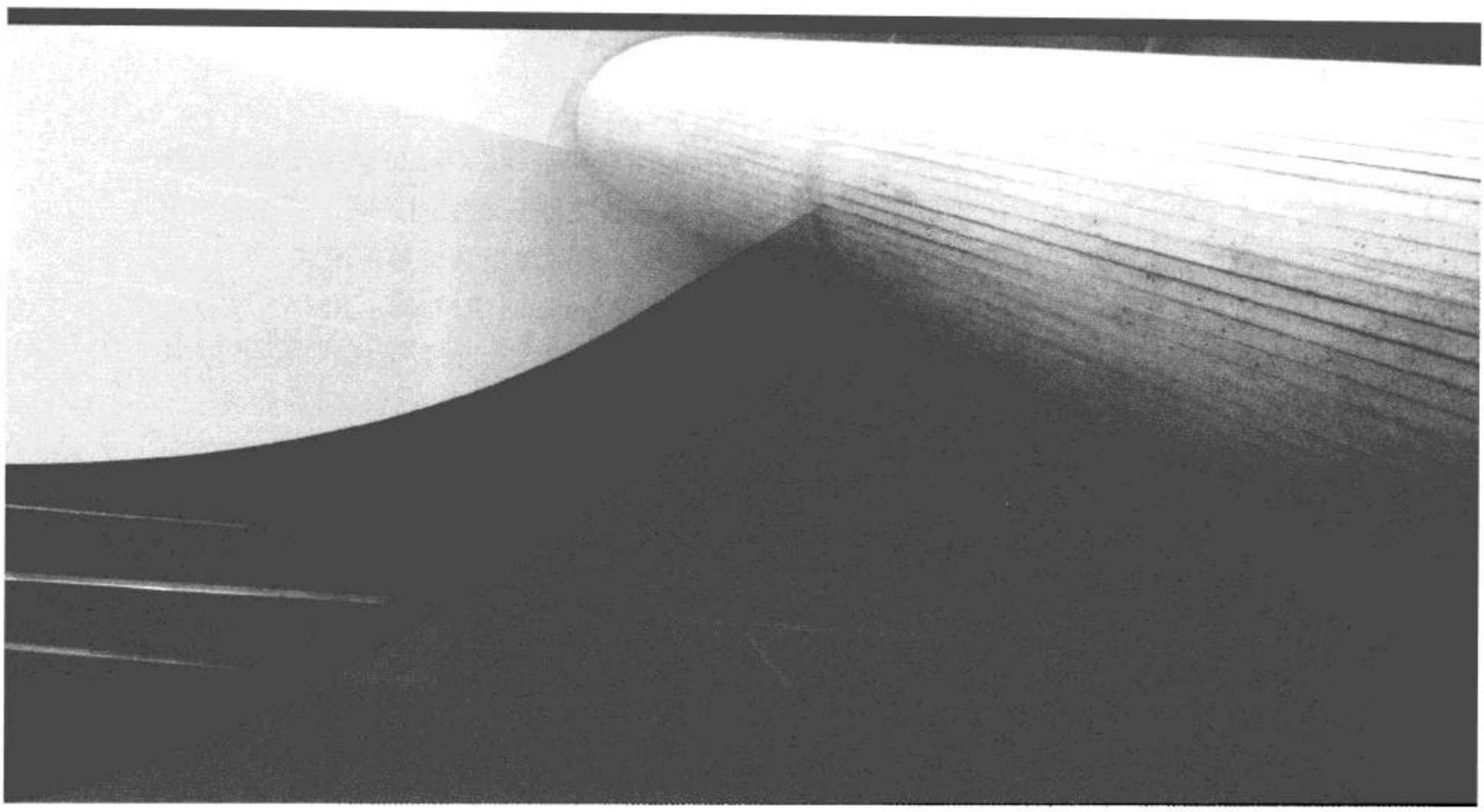
可以看到，Oracle 数据库的备份和恢复进程包括许多步骤。因为 DBA 每次都要确认正确执行了备份，所以他们通常会编写一些脚本来实现这个功能。这样会存在一些问题。首先，脚本会中断。当脚本中断时，如果编写该脚本的 DBA 休假或离开了，那么由谁来维护这个脚本呢？其次，我们必须编写脚本来跟踪添加或删除数据文件，或者必须从所需的脚本中手动添加或删除数据文件。

使用 RMAN 可以得到一个备份和恢复产品，这个产品免费地包含在基本数据库产品中，它将简化备份和恢复进程的复杂性。在遇到问题时，Oracle 也会提供支持。另外，使用 RMAN 可以得到其他备份和恢复进程无法提供的功能。我们将在后面的章节中介绍这些功能。

RMAN 解决了上面提出的所有问题，并且为 DBA 提供了更加有用的其他功能。本书将介绍这些功能，并介绍如何利用这些功能更加方便地执行各种操作以及更加可靠地备份数据库。

1.9　小结

虽然本章没有详细介绍 RMAN，但是简要介绍了一些重要的 RMAN 基础知识，后面的章节将继续讨论这些内容。我们介绍了基础性的备份和恢复概念(如高可用性与备份和恢复计划)，这些概念描述了 RMAN 的主要目的。随后给出了一些需要熟悉的 Oracle 术语的定义，在后面的章节中将会用到这些术语。接下来还回顾了 Oracle 数据库体系结构和内部操作，并且反复强调在突发情况下恢复数据库时，理解 Oracle 的内部工作原理是非常重要的。最后，讨论了 Oracle 中的手动备份和恢复操作，与后面章节介绍的 RMAN 备份和恢复操作相比，您会发现 RMAN 实质上是一种备份和恢复 Oracle 数据库的简单解决方案。



第2章

RMAN体系结构概述

本章将逐一介绍 RMAN 体系结构中的每个组件，并解释这些组件在成功的 Oracle 数据库备份和恢复中所起的作用，要很好地理解这些内容需要读者具有良好的关于 Oracle RDBMS 体系结构的知识。如果对 Oracle 数据库的各类组件不太了解，应当先阅读第 1 章或其他有关数据库管理的初学者指南。讨论完备份和恢复涉及的不同的组件之后，我们将在磁盘上生成一个简单的备份进程，同时介绍所用到的每个组件。

2.1 服务器管理恢复

第 1 章介绍了 Oracle 体系结构中备份与恢复的原理和实践。Oracle 体系结构会创建和运行一些脚本，这些脚本可以获得文件名，并将这些文件与表空间关联，将表空间置入备份模式，使用一个 OS 实用程序来执行复制操作，最后退出这个备份模式。

不过，本书将介绍如何使用恢复管理器(Recovery Manager，RMAN)。RMAN 实现一种服务器管理恢复(Server-Managed Recovery，SMR)。SMR 是一种数据库执行所需的操作以确保自身备份成功的能力，依赖于 Oracle RDBMS 内核中的内置代码可以完成这一功能。想想看，有谁能比数据库本身更了解数据库结构的呢？

SMR 的功能主要体现在能够减少用户的操作。随着企业复杂程度的增加，以及单个 DBA 所负责的数据库数量的增加，很少有人能够亲自维护数十个甚至数百个单独的脚本。换句话说，随着“网格计算”成为主流的计算方法，人们不再需要亲自仔细检查每个数据库备份的所有细节。相反，要由数据库自身处理备份管理的许多细节，从而使我们从繁琐的维护中解脱出来去从事更重要的工作。当然，RMAN 的某些复杂用法超出了 SMR 所宣称的简易性，这也正是本书介绍的内容。但您在 RMAN 上花费的心血和努力将会得到巨大的回报，这是毫无疑问的。

2.2 RMAN 实用程序

RMAN 是 SMR 的具体实现，它是由 Oracle 提供的。RMAN 是一个独立的应用程序，它负责建立到 Oracle 数据库的客户端连接，以访问数据库内部的备份和恢复数据包。RMAN 的核心是命令解释程序，命令解释程序接受所输入的命令，并将这些命令转换为在数据库上执行的远程进程调用(Remote Procedure Call，RPC)。

需要指出并明确的一点是：RMAN 只做少量的工作。虽然事件的协调非常重要，但是备份和恢复一个数据库的具体工作实际上是由目标数据库上的进程完成的，目标数据库(target database)指的是要备份的数据库。Oracle 数据库的内部数据包可以将 RMAN 中的 PL/SQL 块变为系统调用，这些系统调用既可以写入数据库服务器的磁盘子系统，也可以读取数据库服务器的磁盘子系统。

RMAN 实用程序是 Database Utilities 套件的一部分。Database Utilities 套件是一组命令行形式的实用程序，它包括 Data Pump、SQL*Loader、DBNE.WID 和 dbverify。典型安装 Oracle 时会自动安装 RMAN。RMAN 分企业版和标准版两种，如果您只能使用标准版的 RMAN(而不能使用企业版)，RMAN 就只能为备份分配一个信道。如果执行客户端安装，并且选择的是 Administrator 选项而不是 Runtime client 选项，就会安装 RMAN。

RMAN 实用程序由两部分组成：可执行文件和 recover.bsq 文件。recover.bsq 文件实质上是库文件，可执行文件从 recover.bsq 文件中析取代码来创建在目标数据库上执行的 PL/SQL 调用。recover.bsq 文件是整个操作的中枢。这两个文件始终保持链接，并在逻辑上构成 RMAN 客户端实用程序。需要指出的是，recover.bsq 文件和可执行文件的版本必须相同，否则就不能正常工作。

RMAN 实用程序有一种独特的、有序的、可预测的用法：解释在目标数据库上远程执行的 PL/SQL 调用中的命令。RMAN 的命令语言比较独特，需要具备一定经验才能掌握。实质上，RMAN 实用程序完成了我们需要做的所有备份、还原、恢复和处理备份工作。由可执行解释程序负责解释这些程序命令，并且使这些命令与 recover.bsq 文件中的 PL/SQL 块匹配。RMAN 随后将这些 RPC 传递给数据库，并基于用户的请求收集信息。如果命令要求 I/O 操作(即一条备份命令或一条还原命令)，RMAN 会在返回信息时准备另一个进程块，并将这个进程块传递回目标数据库。这些进程块负责执行 OS 的系统调用以进行指定的读或写操作。

RMAN 与数据库权限

RMAN 需要访问目标数据库上 SYS 模式中存在的各种数据包，还需要具有启动和关闭目标数据库的权限，以便在还原数据库时能够建立目标数据库。因此，RMAN 通常以 sysdba(系统管理员)用户的身份连接到目标数据库。我们不需要像 SQL*Plus 那样指定 RMAN 的身份，由于假定每个目标数据库连接中的 RMAN 都是 sysdba 用户，所以在连接目标数据库时，RMAN 会自动作为 sysdba 用户来连接目标数据库。如下所示：

```
RMAN> connect target sys/password
connected to target database: PROD (DBID=4159396170)
```

如果试图以不具有 sysdba 权限的用户身份来连接目标数据库，RMAN 会给出下面的错误信息：

```
RMAN> connect target /
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
ORA-01031: insufficient privileges
```

这是 RMAN 设置和配置阶段中最常见的错误，它出现在没有作为 dba 组成员登录服务器的时候。这个 OS 组将控制服务器上所有 Oracle 数据库的 sysdba 权限的身份验证。dba 是默认的用户名，但不是必需的(在某些操作系统中会使用不同的默认用户名，而不会强迫使用用户名 dba)。在一般情况下，大多数 Unix 系统中都有一个 Oracle 用户，它是 dba 组的一个成员。这是最初安装 Oracle 软件的用户，而且在最先进的配置中将会建立 sudo 命令；则可以执行‘sudo oracle’功能，即以自己的身份登录，也可以拥有 Oracle 用户的权限。如果您以用户名 Oracle 登录，在 RMAN 中以什么身份连接就变得不重要，您将总是作为 sysdba 用户来连接目标数据库，并且能够访问 SYS 模式，同时具有启动和关闭数据库的能力。在 Windows 平台上，Oracle 会创建一个 ORA_DBA 本地组，并将安

装 Oracle 软件的用户添加到这个本地组中。

如果作为 dba 组成员之外的用户登录并需要使用 RMAN，就必须创建和使用一个口令文件来连接目标数据库。如果要使用一个客户端系统通过网络来连接 RMAN，也需要创建和使用一个口令文件。第 3 章将介绍这些配置的步骤。

2.3 RMAN 备份的网络拓扑结构

RMAN 的客户端/服务器体系结构不可避免地会带来混淆，这与人类空间记忆的方法有一定关系。如果善于这方面的思维，您就能非常轻松地理解 RMAN。但如果对这个问题感到困惑，那么请认真阅读下面的内容。

这种混淆完全基于执行 RMAN 的环境，与之相对应的是备份工作的环境。RMAN 是一个客户端应用程序，它通过 Oracle Net 连接来与目标数据库连接。如果目标数据库位于 ORACLE_HOME 主目录中，运行同样位于这个主目录中的 RMAN 可执行文件，Oracle Net 连接就会是一个本地连接。只要在 Oracle 环境中设置了合适的 ORACLE_SID 变量，就不需要为这个连接提供一个 Oracle Net 别名。否则，我们必须配置 tnsnames.ora 文件，在文件中指定目标数据库，并且在将要运行 RMAN 的位置上完成这个操作。图 2-1 说明了不同 RMAN 位置的网络拓扑结构。

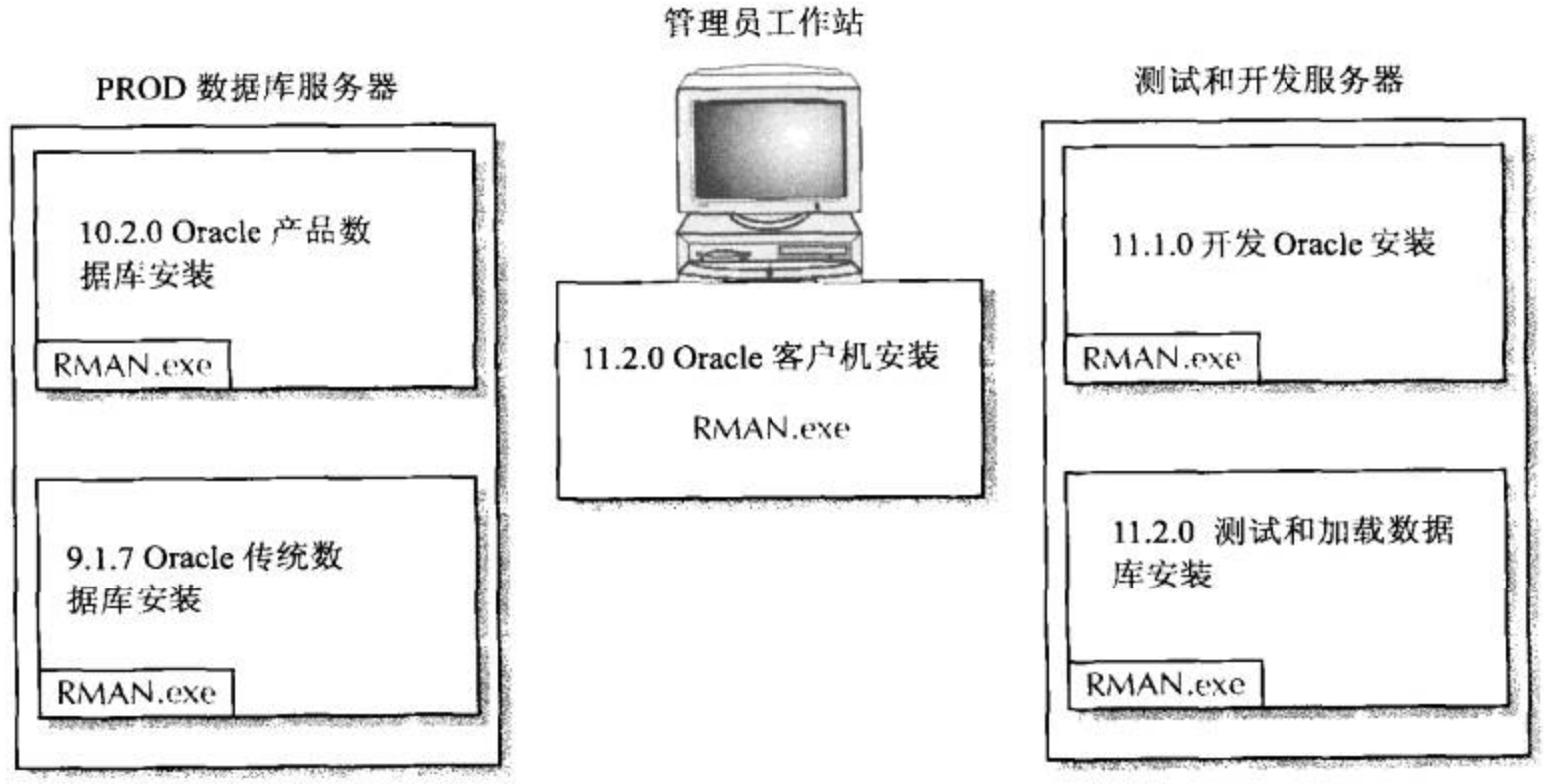


图 2-1 RMAN 可执行文件的 5 种不同位置(版本)

2.3.1 远程运行 RMAN

如果您负责管理遍布企业内部的许多数据库，最好在单个客户端系统上将 RMAN 应用程序合并，这样就可以在这个客户端系统上更好地管理 tnsnames.ora 文件的内容。并

且可以合并所有 RMAN 脚本，还不会在以后因搞不清 RMAN 运行的位置而头疼。您将清楚地知道 RMAN 运行的位置：在膝上型电脑上，在台式电脑上，或者是在 Linux 工作站上。如果在 RMAN 配置中使用了恢复目录(recovery catalog)，由于每次操作 RMAN 时都将生成一个以上的 Oracle Net 连接，因此这样的客户端/服务器模型才能让您不至于混淆。另一方面，运行与目标数据库不同系统(或者是不同的 ORACLE_HOME 主目录)上的 RMAN 时，我们需要建立一个口令文件，这样在每个目标数据库上都需要更多的配置和管理。

谁需要使用恢复目录？

恢复目录用于存储 RMAN 备份的历史记录，它含有有关备份发生时间、备份内容和备份大小的元数据。历史记录中还含有恢复备份所需的关键信息。元数据是从默认位置(目标数据库控制文件)析取的，并保留在用户模式的数据库表内。是否一定需要使用恢复目录呢？答案是否定的——只有存储的脚本功能实际需要该目录时才使用。

如果最终要使用 standby 配置(第 20 章将介绍)或 sync/split 配置(第 22 章将介绍)来配置更为复杂的环境，将需要使用恢复目录。

恢复目录会带来方便吗？答案是肯定的。恢复目录是否增加了复杂性？答案也是肯定的。第 3 章将讨论恢复目录的创建和设置，并详细解释是否使用恢复目录的原因。本章稍后部分会介绍恢复目录体系结构。

如果要建立一个 RMAN 和目标数据库之间的远程连接，则需要创建一个使用专用服务器进程连接到目标数据库的 tnsnames.ora 条目。RMAN 不能使用共享服务器(通常称为多线程服务器或 MTS)来建立数据库连接。因此，如果使用的是共享服务器(共享服务器是所有新安装的默认设置)，就需要创建一个使用专用服务器进程的 Oracle Net 别名。从下面的 tnsnames.ora 文件中可以看出这两种情况的差异。需要注意的是第一个别名条目用于专用服务器进程，第二个条目则使用共享服务器体系结构。

```
PROD_RMAN =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = cervantes)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = prod)
    )
  )
PROD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = cervantes)(PORT = 1521))
    )
    (CONNECT_DATA =
```

```
        (SERVER = SHARED)
        (SERVICE_NAME = prod)
    )
)
```

2.3.2　在目标数据库的 ORACLE_HOME 中本地运行 RMAN

从每个目标数据库中本地运行 RMAN 实际上是管理具有数百个(或数千个)数据库目标的大型企业的唯一方法。因此 RMAN 一直具有兼容性问题，将 rman.exe 与目标数据库紧密绑定可在长期运行中节省时间。虽然以这种方式部署 RMAN 备份存在一些缺陷，但根据我们多年以来的部署经验，我们认为这是最佳的方式。

本地运行 RMAN 意味着只需本地连接数据库，而不需要建立口令文件和配置 tnsnames.ora 文件。必须牢记：这种方式的简易性同时也正是它的缺陷，一旦要引入恢复目录或者执行数据库复制操作，就必然会引入我们先前设法避免的那些问题。这种方式在使用中也会产生混淆，由于我们通常在本地连接数据库，所以很有可能会连接到错误的目标数据库。同样，这种方式也往往会让我们不明白究竟是从哪个环境连接的。如果系统中安装了多个 Oracle 软件，您就会陷入困境。可能会认定连接的是 PROD 实例，而实际上设置的却是用于 TEST 实例的 ORACLE_HOME 和 ORACLE_SID 环境变量。

从桌面工作站中运行 RMAN 和在每个目标数据库服务器中本地运行 RMAN 的真正区别可能在于 OS 主机安全性。为了本地运行 RMAN，我们总是需要能够作为 OS 级别上的 oracle 用户登录到每个数据库服务器，并且具有为该用户定义的权限。然而，如果总是从远程 RMAN 可执行文件中建立数据库的 Oracle Net 连接，则永远不需要有主机登录凭证。

因此，一定要灵活地选择运行 RMAN 的方式。我们已经声明过首选的方法，接下来就要告诉您该方法的弊端。即使我们的设计被简化为客户端 RMAN 或服务器 RMAN，还是应该采用图 2-2 提供的混合模型来满足设计要求。图 2-2 中存在 5 种不同的运行情况。

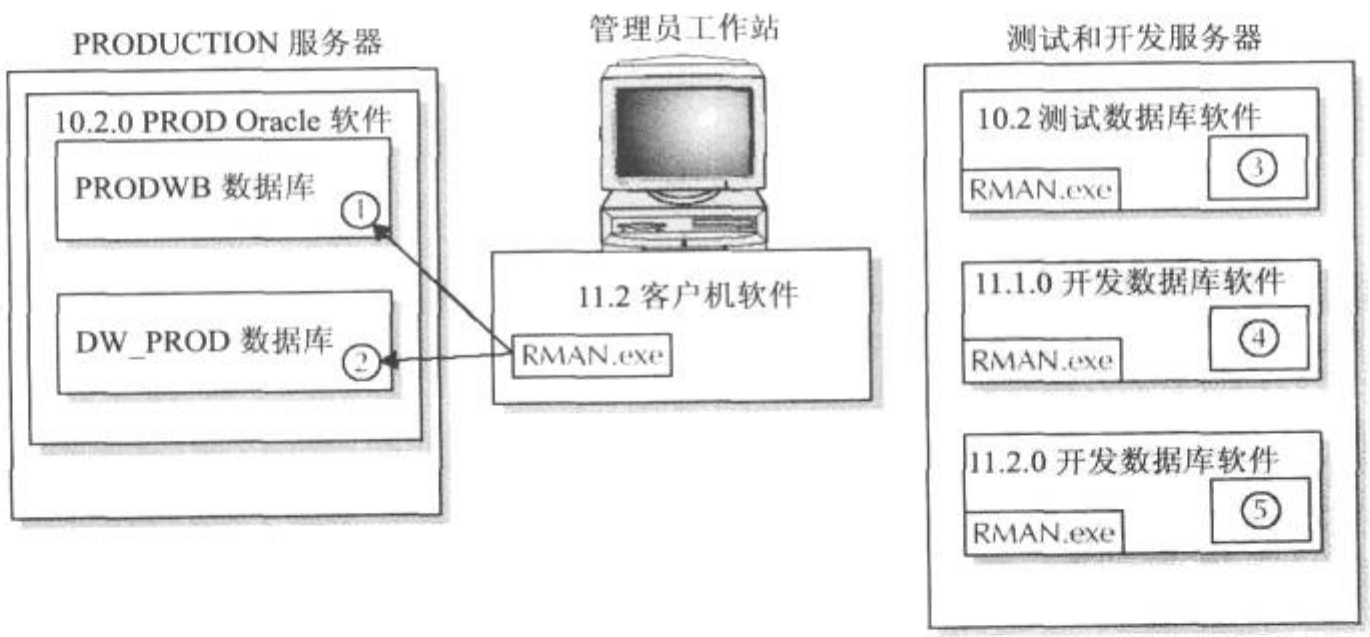


图 2-2　在企业中运行不同版本的 RMAN 可执行文件

(1) RMAN 将作为 DBA 工作站上的一个客户端连接来运行，因为负责备份 PRODWB 和 DW_PROD 的 DBA 没有产品数据库服务器上的 oracle 用户口令。

(2) RMAN 将远程备份 DW_PROD，和 PRODWB 一样，这是由于在数据库产品服务器上的安全限制。

(3) 10.2 TEST 数据库是通过从 TEST $ORACLE_HOME 运行的本地 RMAN 可执行文件来备份的。

(4) 本地执行 11.1.0 DEV 数据库备份。因为 DBA 具有测试和开发服务器上的 oracle 用户权限，这是切实可行的方法，并且它最小化在本地工作站上维护的客户端安装数量。

(5) 出于与 11.1.0 DEV 数据库相同的原因，也在本地备份 11.2.0 DEV 数据库。

一定要维持 RMAN 拓扑结构的灵活性，这样既可以在 NOCATALOG 模式中使用本地 RMAN 可执行文件来执行备份工作，也可以在其他情况下执行远程 RMAN 作业。

2.4　数据库控制文件

到目前为止，我们已经讨论了 RMAN 可执行文件及其在 Oracle 11g 的服务器管理恢复进程中的作用，并且指出实际工作是在目标数据库上完成的，即目标数据库备份本身。接下来必须要讨论控制文件在 RMAN 备份或恢复进程中的作用。

控制文件平常的作用是负责数据库的物理结构。顾名思义，控制文件控制数据库查找物理文件的位置，并控制每个文件当前包含(或应当包含)的头信息，其内容包括数据文件信息、重做日志信息和归档日志信息。控制文件还含有与数据库关联的关键文件的每一个文件头的快照(snapshot)。正是因为包含有丰富的信息，所以在 RMAN 出现之前的任何恢复操作中，控制文件一直是主要的组件(第 1 章已经详细介绍了这个问题)。

由于控制文件存储的是数据库文件信息，所以 RMAN 会利用控制文件获取备份所需的信息，具体的操作方式是：RMAN 使用控制文件来编译文件列表，得到检查点信息，并确定可恢复性。通过直接访问控制文件，RMAN 可以编译文件列表而不需要用户创建文件列表，这样避免了备份脚本时的一个非常繁琐的步骤，并且在添加一个新文件时不需要更改这个脚本。控制文件会存储新文件的信息，因此 RMAN 也能从控制文件中得到这个信息。

控制文件还能用作 RMAN 目录。RMAN 完成数据库任何部分的备份后，会在控制文件中添加该备份的记录，以及说明备份开始和结束时间的检查点信息。这也是从 Oracle 7 到 Oracle 8 版本控制文件大小呈指数增长的一个主要原因，即控制文件中存在 RMAN 表。控制文件中的这些记录通常称为元数据，它与实际备份中记录的数据有关，它也可以存储在恢复目录中(参阅第 3 章)。

2.4.1　在控制文件中重用记录

控制文件可以增长以满足空间增长的需求。例如，当添加一条用于新数据文件、新日志文件或新 RMAN 备份的记录时，控制文件会通过扩展来解决增加空间的问题。不过，控制文件的扩展有一定的限制。通常，大多数数据库会存在许多年，成千上万的重做日

志切换和检查点都在它上面运行。对此，控制文件必须能够删除某些不再需要的数据。这样，在需要空间时，控制文件会根据时间来删除信息，并以轮询(round_robin)方式查找并重用表中的某些空间。当然，有些信息是不能被删除的。例如，数据文件列表信息对于每时每刻的数据库操作来说至关重要，这类新记录必须有足够的新空间可用。

控制文件将其内部数据记录分为两类：循环重用记录和非循环重用记录。循环重用记录包含可以从控制文件中删除的信息。例如，归档日志历史信息可以被删除，并且不会影响产品数据库。非循环重用记录是那些不能被删除的记录，如果控制文件因这类记录而空间不足，则会删除这些记录以得到更多的空间。非循环重用记录包括数据文件列表和日志文件列表。

控制文件中的 RMAN 备份记录属于循环重用记录类别，如果含有这些记录的控制文件区域已满，这些记录就会被删除。这会给恢复带来灾难，致使控制文件中不存在任何备份记录，就好像从未发生过备份一样。一定要记住：如果控制文件不含有任何 RMAN 备份记录，RMAN 就不能使用备份来执行恢复操作(第 12 章将介绍如何重新向控制文件记录中添加备份)。因此，控制文件是 RMAN 的一个关键因素，没有备份记录，我们就不能执行恢复操作。如果删除某些备份记录，我们就不能再使用这些记录指向的备份。

但是，这种情况您不必担心。删除记录并不可怕，它只发生在控制文件空间已满的情况下，因此被删除的备份肯定是非常陈旧的。我们还可以设置更大的时间帧(timeframe)来决定控制文件删除记录的时间。时间帧由 init.ora 文件的 CONTROL FILE_RECORD_KEEP_TIME 参数控制，该参数的默认值为 7(以天为单位)，这表明如果记录的保留时间少于 7 天，就不会删除该记录，此时只能扩展控制文件区域。可以设置更大的参数值，如 30(天)，这样就会经常扩展控制文件，且只有当记录的保留时间超过一个月时才会重写这些记录。一般来说，设置该参数为较大的天数值比较好。参数值被设为 0 时将不会发生重用记录(不过我们不建议这种设置)，相应地控制文件将不断扩展直至无法管理。

此外，如果要实现一个恢复目录，不必担心会循环重用记录。因为只要您在 CONTROL FILE_ RECORD_KEEP_TIME 参数指定的时间内至少实现一次恢复目录同步，那么这些记录将随着时间的推移被删除——恢复目录则从来不会删除记录。

重新构建控制文件(RMAN 用户必读)

通常在某些条件下需要重新构建数据库控制文件，如重新设置 MAXLOGFILES 参数或 MAXLOGHISTORY 参数。由于某些参数定义了保存非循环重用记录的内部控制文件表的大小，所以只有在重新构建控制文件时才能设置这些参数。如果需要更大的内部控制文件表，就必须重新构建控制文件。

如果使用 RMAN 而不使用恢复目录，就必须重视控制文件的重新构建。执行

```
alter database backup control file to trace;
```

命令时，生成的脚本不包含控制文件中标识备份的信息。如果不存在备份记录，就不能在使用这些备份进行恢复时访问它们，此时所有的 RMAN 信息丢失并且无法恢复。在重新构建控制文件时，只有那些用 RMAN 设置的永久配置参数才被重新构建。在 Oracle 10g 中有一些新的机制可用于生成控制文件中的有限备份元数据，但仍然需要许多手动

操作，并且无法完全生成所有的元数据。因此我们建议无论如何要避免重新构建控制文件。

如果将控制文件备份为一个二进制文件而不是一个跟踪文件，所有备份信息都将被保留。该命令类似于：

```
alter database backup controlfile to '/u01/backup/bkup_cfile.ctl';
```

2.4.2　快照控制文件

可以说控制文件是一个非常繁忙的文件，它存储数据库的结构信息，这些信息包括用于恢复的检查点 SCN 信息。连续的 SCN 和文件管理对于数据库的生命周期来说至关重要，因此 RDBMS 必须能够持续地使用控制文件。

这样会为 RMAN 带来一个问题。RMAN 开始备份每个数据文件时需要得到一个一致的控制文件视图，此时 RMAN 只需要知道备份开始时的最新的检查点信息和文件结构信息。开始备份后，RMAN 需要这些信息在备份操作期间保持一致，也就是说 RMAN 需要一个读取一致的控制文件视图。除非 RMAN 在备份持续时间内锁定控制文件，否则数据库会不断更新控制文件，所以这是不可能的。但是，锁定控制文件意味着数据库不能执行检查点操作和切换日志，或者不能产生新的归档日志，这些操作是不可能的。

RMAN 使用快照控制文件(snapshot controlfile)来解决前面提出的问题，快照控制文件是控制文件的副本。RMAN 只在备份和同步操作期间使用快照控制文件。这些操作开始时，RMAN 会根据实际控制文件内容来刷新快照控制文件，这样会短暂地锁定控制文件；随后，RMAN 会切换到快照并在备份期间持续使用这个快照。这种方式具有读取一致性，且不妨碍数据库活动。

在默认情况下，快照控制文件位于 Unix 平台上的 ORACLE_HOME/dbs 目录中，或者是在 Windows 的 ORACLE_HOME/database 目录中。快照控制文件的默认名为 SNCF<ORACLE_SID>.ORA。使用 configure snapshot controlfile 命令可以在任何时候更改快照控制文件名：

```
configure snapshot controlfile name to '<location\file_name>';
```

在某些情况下可能出现以下有关快照控制文件的错误，一般人直到这时才会知道还有这个文件：

```
RMAN-08512: waiting for snapshot controlfile enqueue
```

如果快照控制文件头不是被请求插入队列的进程锁定而是被另一个进程锁定时，就会出现这样的错误。如果执行多个备份工作，就有可能同时从两个不同的 RMAN 会话进程执行两个备份操作。要修正这个错误，可以打开 SQL*Plus 会话并运行下面的 SQL 语句。

```
SELECT s.sid, username AS "User", program, module, action, logon_time"Logon",
    l.*
```

```
FROM v$session s, v$enqueue_lock l
WHERE l.sid = s.sid and l.type = 'CF' AND l.id1 = 0 and l.id2 = 2;
```

2.5 RMAN 服务器进程

RMAN 生成一个到目标数据库的客户端连接，并创建两个服务器进程。其中的主要进程生成对 SYS 模式中的数据包的调用，从而执行备份和恢复操作，该进程会在备份和还原期间协调信道进程的工作。

另外一个次要进程(或影子进程)轮询 RMAN 中的所有长事务(long-running transaction)并在内部记录信息。在 V$SESSION_LONGOPS 视图中可以查看到下面的轮询结果：

```
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
       ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"
FROM V$SESSION_LONGOPS
WHERE OPNAME LIKE 'RMAN%'
AND OPNAME NOT LIKE '%aggregate%'
AND TOTALWORK != 0
AND SOFAR <> TOTALWORK
/
```

还可以在 V$SESSION 视图中查看这些进程。当 RMAN 分配一个信道时，它会在输出中提供如下所示的会话 ID 信息：

```
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=16 devtype=DISK
```

“sid”信息对应于 V$SESSION 中的 SID 列。因此，我们可以构造下面的查询：

```
SQL> column client_info format a30
SQL> column program format a15
SQL> select sid, saddr, paddr, program, client_info
    from v$session where sid=16;
      SID SADDR    PADDR      PROGRAM        CLIENT_INFO
--------- -------- ---------- -------------- ------------------------
       16 682144E8 681E82BC   RMAN.EXE       rman channel=ORA_DISK_1
```

2.5.1 RMAN 信道进程

除了默认的两个进程之外，在备份或还原操作期间还会为分配的每个信道单独创建一个进程。RMAN 将信道视为目标数据库上的服务器进程，该进程在备份期间协调读取数据文件和写入指定位置的操作，在还原期间则协调读取备份位置和在数据文件位置写入数据块的操作。信道只有两种类型：磁盘信道和磁带信道。我们不能为一个备份操作同时分配这两种信道，备份只能写入磁盘或写入磁带。与后台 RMAN 进程一样，可以从数据字典(data dictionary)跟踪信道进程，并且将信道进程在 OS 级别(OS level)上与一个

SID 关联。正是这些信道进程的活动被轮询的影子进程记录到 V$SESSION_LONGOPS 视图中。

2.5.2　RMAN 和 I/O 从属

如果在目标数据库上配置了 I/O 从属，RMAN 就可以利用 I/O 从属。为了使用 RMAN 备份和还原，有两种可用的 I/O 从属：磁盘 I/O 从属和磁带 I/O 从属。

使用 DBWR_IO_SLAVES 参数可以配置磁盘 I/O 从属。这个参数可以被设置为任意数值，它的主要作用是在将“脏”缓冲区(即写满的缓冲区)的内容写入磁盘时唤醒另外的 DBWR 从属来执行磁盘写操作。如果该参数被设置为非 0 值(如 1、12 或 32)，RMAN 会自动在每个信道中切换使用 4 个 I/O 从属来帮助将数据块读入 RMAN 内存缓冲区。这是一个不错的功能，但是 RMAN 分配内存的方式有很大的变化。只有在 OS 平台不支持本地异步 I/O 或者禁止 Oracle RDBMS 使用异步 I/O 时，DBWR_IO_SLAVES 参数才会起到重要的作用。如果支持异步 I/O，就不需要使用磁盘 I/O 从属。

磁带 I/O 从属可帮助服务器进程访问磁带设备。如果 BACKUP_TAPE_IO_SLAVES 参数被设置为 TRUE，RMAN 会为每个磁带信道进程分配一个 I/O 从属来帮助在磁带位置上执行写操作。与磁盘 I/O 从属不同的是，BACKUP_TAPE_IO_SLAVES 参数只对 RMAN 磁带备份起作用。由于磁带设备不存在本地异步 I/O，我们建议将该参数设置为 TRUE，这有助于保持磁带设备传输，以达到更佳的备份和还原性能。第 16 章将详细讨论磁带传输问题。

2.6　RMAN 使用的 SYS 数据包

RMAN 服务器进程在协调信道的工作时要访问两个 SYS 数据包：DBMS_RCVMAN 数据包和 DBMS_BACKUP_RESTORE 数据包，这两个数据包包含了目标数据库中 RMAN 的所有功能。

2.6.1　SYS.DBMS_RCVMAN 数据包

DBMS_RCVMAN 数据包用来访问控制文件中的表并将此信息传递给 RMAN，使 RMAN 可以构建准确反映数据库结构的备份与还原操作。DBMS_RCVMAN 数据包负责在执行任何操作前设置 TIME 运算符和校验数据文件头中的检查点信息，并检查文件位置和大小，以及有关节点相似性(在 RAC 环境中)和磁盘相似性的其他信息。这种信息会影响 RMAN 的性能，并且 RMAN 在构建实际备份/还原命令之前自动执行负载均衡(load-balancing)算法和性能增强(performance-enhancing)算法。这些将在第 16 章详细介绍。

2.6.2　SYS.DBMS_BACKUP_RESTORE 数据包

SYS.DBMS_RCVMAN 数据包访问控制文件并校验所有必需的信息。这些必需的信息被传递回 RMAN 服务器进程，RMAN 服务器进程随后会基于 recover.bsq 文件中的代码创建一些 PL/SQL 块。这些 PL/SQL 块含有对 DBMS_BACKUP_RESTORE 数据包的调

用，实际上 RMAN 中是由 DBMS_BACKUP_RESTORE 数据包来创建系统调用以备份数据文件、控制文件和归档的重做日志。RMAN 接收从 DBMS_RCVMAN 数据包返回的信息，并在信道间基于负载均衡算法分配工作，然后创建一系列 DBMS_BACKUP_RESTORE 数据包调用。

在 V$SESSION_LONGOPS 视图中可以跟踪的是 DBMS_BACKUP_RESTORE 数据包的工作。它执行备份和还原操作，此外还可以以受限方式访问控制文件。DBMS_BACKUP_RESTORE 数据包访问控制文件以备份该文件(实际上备份的是快照控制文件)，在备份完成后写入备份信息。一旦完成了一个备份集，DBMS_BACKUP_RESTORE 数据包就会在控制文件的表中写入备份开始时间、备份时间、备份大小和备份名的信息。

内核中的 RMAN 数据包

创建数据库时，运行 catproc.sql 脚本可以默认安装上述两个 RMAN 数据包。数据库创建期间无法跳过这一步骤，因此在 8.0.3 版本以后的每个 Oracle 数据库中都存在这些 RMAN 数据包。这样就不必配置 RMAN，我们可以立即运行 RMAN 并开始备份数据库。

这些 RMAN 数据包还具有另一个重要的特性：它们被硬编码到 Oracle 软件的库文件中，因此即使没有打开数据库也可以调用这些数据包。我们知道，大多数数据包只在数据库打开时才能被使用。不过，RMAN 可以在数据库实例处于未加载模式或加载模式时编写对 DBMS_BACKUP_RESTORE 数据包的调用。这是一个关键要素，其原因显而易见：需要在没有打开数据库时也能够备份和还原这个数据库。

您会注意到这样的问题：如果要使用 RMAN 连接目标数据库，目标数据库应该处于什么状态？需要启动实例吗？数据库需要加载吗？还是必须打开？实际上 RMAN 可以连接这 3 种状态的目标数据库，但是目标数据库必须至少处于未加载模式下。

2.7 备份数据块

正如第 1 章所述，即使使用高级的备份技术，备份的单元仍然是数据文件。最终进行备份的 OS 实用程序将查看并备份整个文件，因此需要特别注意保证 Oracle 数据块的完整性。然而，RMAN 与之不同，因为 RMAN 是集成在 RDBMS 中的，它是在数据块层面访问数据，这与数据库本身所使用的方式相同。

块级别的访问是 RMAN 区别于其他备份工具之处，也许您现在对这一点还不太明白，不过本书将帮助您逐步了解 RMAN 的备份策略。正是由于 RMAN 采用了这一层面的访问，才使得它具有诸多优势，因此我们可以更有效地利用数据块进行备份和恢复。

2.7.1 数据块备份概述

数据块备份的工作原理是：RMAN 基于备份算法规则来编译要备份的文件列表。基于信道数和同时备份的文件数，RMAN 在 Oracle 共享内存段中创建一些内存缓冲区。一般是在 PGA 中，不过有时内存缓冲区会被推入 SGA。信道服务器进程随后开始读取数据文件，并在 RMAN 缓冲区中填充这些数据块。一个缓冲区被填满时，输入缓冲区中的

数据块会被推入输出缓冲区。数据文件中的每个数据块都会发生这种内存对内存的写操作(memory-to-memory write)。如果数据块符合备份的标准并且内存对内存的写操作没有检测到损坏(corruption)，则该数据块会保留在输出缓冲区中直至输出缓冲区被填满。一旦输出缓冲区被填满，输出缓冲区中的内容就会被推至备份位置(可能是磁盘或磁带)。

一旦所有文件已通过内存缓冲区过滤，就完成了备份片，同时 RMAN 会在目标数据库的控制文件中写入备份片的完成时间和备份片的名称。

2.7.2　块级备份的优点

每个数据块从磁盘移入内存时都会发生内存对内存的写操作，在这个操作期间会检查数据块的损坏。损坏检查是 RMAN 的一个重要特性，第 12 章将详细介绍这个问题。需要记住的是，执行代理复制操作时不会使用数据块检查。

空块压缩

访问数据块时可以选择空块压缩(null block compression)，我们可以删除从来没有使用过的数据块(头为 0)，并在执行内存对内存的写操作时丢弃这些数据块。因此，我们只需要备份使用过的数据块，从而执行更有效的备份操作。

这里适合介绍与空块压缩有关的各种误解。第一个误解是要删除空数据块。RMAN 到数据库的两个接入点是文件头和数据块头。RMAN 只能从数据块头信息或文件头信息中推断该数据块的内容。为什么不需要空间管理信息呢？事实上，空间管理信息只在数据库打开时有效，而 RMAN 不依赖于打开的数据库。我们必须只依赖于无需打开数据库就能获得的信息，也就是文件头和数据块头。因此如果截断一个表，表中所有存储过信息的空数据块都会备份，因为 RMAN 只知道内存段初始化了这些数据块，而不知道这些数据块为空。

第二个与空块压缩有关的常见误解是备份期间的增量备份可以节省备份的时间。对特定的情况(备份设备是一个大瓶颈)来说，这种说法在某种程度上是正确的。如果到备份位置(磁盘或磁带)的数据流很快，那么删除内存中数据块的动作不会节省多少时间。这是因为 RMAN 仍会将文件中的每个数据块都读入内存，它只是不将这些数据块都写入到输出设备。即使是在增量备份期间(增量备份基于增量检查点 SCN 来删除数据块)，我们仍然要通过检查每个数据块头来观察数据块自前一次增量备份以来是否被更改过。增量备份可以节省备份位置的空间，并提供更快的恢复形式，但是这不意味在实际的备份期间，增量备份可以有效地节省时间。

未使用的块压缩

未使用的块压缩是忽略未使用块的第二种机制。它与空块压缩的结果匹配：即，未初始化的块不会被备份。然而，在 10.2.0.3 版本之后，即使是使用过的块，如果是空块，也不会被备份。这种方法减少了备份的时间，因为它访问空间管理信息并检查每个段的位图，从上述的信息中构建初始化的块的列表，不会对未初始化的块进行备份。这意味着一般不会将所有数据读入到内存中，由此节省了备份时间。

未使用的块压缩可以自动使用，但是它不能在数据库中所有的块中使用。这种方法有体系结构上的限制，并且它需要满足如下条件：

- 要求备份是完全备份或是 0 级增量备份。
- 备份是磁盘备份(或是 Oracle 安全备份)。
- COMPATIBLE 的初始化参数设置为 10.2 或是更高的值。
- 待备份的数据库没有可靠的还原点。
- 数据文件是本地管理的(也就是说，空间管理信息保存在文件头中，而不是保存在数据字典中)。

最后一个条件，数据文件的本地管理，允许 RMAN 获得成功的未使用的块压缩的位图信息，该方法不需要两次访问(可能不可用的)数据字典。

二进制压缩

在 10g 版本中，RMAN 最终使用空白压缩，如同 ZIP 实用程序所完成的那样。这就提供了备份块自身的实际压缩。此外，新的块更改跟踪文件允许 RMAN 在备份期间跳过一些数据块，不将它们读入内存缓冲区，因此如果启动更改跟踪，增量备份就可以节省时间。关于压缩和块更改跟踪的更多细节，请查看第 9 章的完整介绍。

在 11gR2 中，可以使用 Oracle 高级压缩，它提供了三个不同级别的压缩方法，可以让二进制压缩与您所在的环境相匹配。三个级别分别为高级、中级和低级。高级，用在限制带宽的环境中，在这种环境中网络资源的限制访问权是优先的；中级，用在对 CPU 利用的压缩率的合并；低级，CPU 的使用成为网络带宽或备份片大小的限制因素。

块级备份的备份性能

块级备份从生成重做的角度也提供性能增益。正如第 1 章所介绍的，如果使用老式的热备份方法，则运行热备份模式的表空间时生成的重做数量有时会呈指数增长。这会造成过多的重做日志切换、检查点故障和生成巨量的归档日志，巨量的归档日志会给日志归档位置上的空间管理带来一定困难。

另一方面，由于在备份期间不要求保证数据块的一致性，因此 RMAN 不要求使用热备份模式。RMAN 对数据块的访问允许 RMAN 与写“脏”缓冲区的 DBWR 进程协调，并且在将这个数据块读入内存以前它可以等待直至数据块达到一致。因此，数据块不会被转储为重做，并且备份中的数据块总是一致的。

当然，RMAN 也不要求使用 ARCHIVELOG 模式。事实上，RMAN 不允许我们在数据库打开时备份一个数据文件，除非是处于 ARCHIVELOG 模式，否则会给出下面的错误信息：

```
ORA-19602: cannot backup or copy active file in NOARCHIVELOG mode
```

RMAN 可以利用块级备份提供容易被忽略但及其有用的恢复选项：块介质恢复(block media recovery)。现在，如果出现错误“ora-1578: block corruption detected(检测到数据块损坏)”，不需要恢复整个文件以及执行恢复操作，RMAN 可以简单地恢复坏数据块并执

行恢复操作，也就是说数据文件中的其他数据在恢复期间仍然可用。第 12 章将进一步讨论这个问题。

上面简单介绍了块级备份的要点。使用 RMAN 进行块级备份会带来很大的好处，本书后面部分会大量介绍使用块级备份的优点。

2.8　内存中的 RMAN

RMAN 会在内存中构建一些缓冲区，然后通过这些缓冲区将数据块写入到备份中。内存的利用与 PGA(有时是 SGA)的总体大小有关。内存缓冲区分为输入缓冲区和输出缓冲区。输入缓冲区(input buffer)填充从备份文件中读取的数据块；输出缓冲区(output buffer)则在执行内存对内存的写操作时填充需要备份的数据块，一旦输出缓冲区被填满，输出缓冲区的内容就会被写到备份位置。在磁盘或磁带上备份与从磁盘或磁带上还原相比，内存缓冲区之间存在不同。图 2-3 示例了内存中的输入缓冲区和输出缓冲区的分配，并说明了如何将两个数据文件的备份多路复用在一个备份集中。

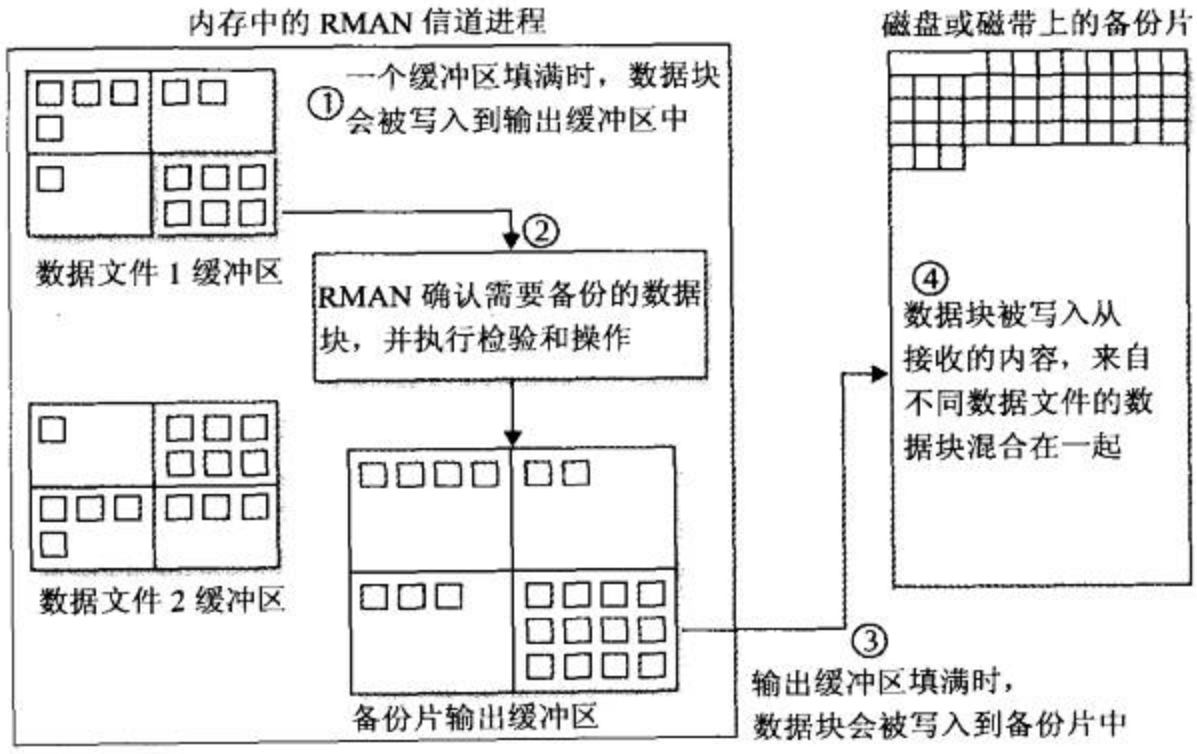


图 2-3　内存中的输入缓冲区和输出缓冲区

2.8.1　输入内存缓冲区

备份数据库时，输入内存缓冲区的大小和数目取决于实际执行的备份命令，事实上它主要取决于在一个备份中多路复用(multiplexing)的文件数。多路复用指的是在同一个备份片中备份其数据块的文件数。为了保持合理范围的内存分配，根据一起备份的文件数，分配内存缓冲大小时需要应用下面的规则。

- 如果备份集内的文件数小于或等于 4 个，则 RMAN 会为每个文件分配 4 个大小为 1MB 的缓冲区。缓冲区总和小于或等于 16MB。

- 如果备份集内的文件数多于 4 个但少于或等于 8 个，则 RMAN 会为每个文件分配 4 个大小为 512KB 的缓冲区。缓冲区总和确保少于或等于 16MB。
- 如果多路复用的文件数多于 8 个，则 RMAN 会为每个文件分配 4 个大小为 128KB 的缓冲区。这就确保每个要备份的文件占用 512KB 的缓冲区内存。

一定要记住要在每个信道基础上分配内存大小。因此，如果分配两个信道来备份具有 32 个数据文件的数据库，RMAN 会根据负载均衡来分配两个信道，而不是为每个信道分配 16 个文件。如果某些文件比其他的文件大很多，我们可能在一个备份集内备份 8 个文件，在另一个备份集内备份 24 个文件。如果是这样，则用于第一个信道(8 个文件)的缓冲区会分配 16MB 的输入缓冲区(8 个文件，每个文件有 4 个缓冲区，每个缓冲区为 512KB)，用于第二个信道(24 个文件)的缓冲区会分配 12MB 的输入缓冲区(24 个文件，每个文件的缓冲区为 512KB)。

可以使用如下的查询来监控备份运行时基于每个文件的缓冲区大小：

```
SELECT set_count, device_type, type, filename,
buffer_size, buffer_count, open_time, close_time
FROM v$backup_async_io
ORDER BY set_count,type, open_time, close_time;
```

1. 备份到磁盘时的输出内存缓冲区

除了输入缓冲区之外，RMAN 还会根据输出设备分配输出缓冲区。如果备份到磁盘，则 RMAN 将分配输出缓冲区以在数据溢出到备份片之前接受来自输入缓冲区的数据块。此时，每个信道有 4 个输出缓冲区，每个输出缓冲区的大小为 1MB，因此每个信道的内存区域通常为 4MB。

2. 备份到磁带时的输出内存缓冲区

备份到磁带时的内存分配是不同的，这是由于磁带设备的 I/O 速率较慢。在磁带上备份或从磁带上还原时，RMAN 会为每个信道进程分配 4 个输出缓冲区，每个输出缓冲区的大小为 256KB，因此每个信道的内存区域通常为 1MB。

2.8.2 用于还原的内存缓冲区

还原操作期间的内存利用与备份操作期间的内存利用稍有不同，这是因为它们的作用是相反的：备份操作是读取数据文件并写入备份位置，而还原操作则是读取备份位置并写入数据文件。还原磁盘备份时，输入缓冲区的大小为 1MB，同时 RMAN 会为每个信道分配 4 个缓冲区。还原磁带备份时，RMAN 会分配 4 个输入缓冲区，每个缓冲区的大小等于 BLKSIZE 参数的值(默认值为 256KB)。用于还原的输出缓冲区通常为 128KB，并且每个信道有 4 个输出缓冲区。

多节备份和存储器

在 11g 中，Oracle 引进了一个新的功能，允许 RMAN 使用多信道备份一个单一的大

容量文件。这意味着先前的存储器的内存输入/输出缓冲区会话仍然有效，但缓冲区是按信道备份的，而不需要按文件备份。以此方式，每个信道为将要备份的文件的每个部分打开四个输入缓冲区。输出缓冲区按备份片保留与前述算法一致。

2.8.3　RMAN 内存利用：PGA 与 SGA

在磁盘上的备份会使用 PGA 内存空间作为备份缓冲区，PGA 内存空间从用于信道进程的内存空间中分配。如果操作系统没有配置本地异步 I/O，则可以利用 DBWR_IO_SLAVES 参数使用 I/O 从属来填充内存中的输入缓冲区。如果设置 DBWR_IO_SLAVES 参数为任意的非零值，则 RMAN 会自动分配 4 个 I/O 从属来协调输入缓冲区内存中的数据块加载。为了实现这一功能，RMAN 必须利用一个共享内存区域。因此，用于磁盘备份的内存缓冲区会被推入共享池，如果存在大池，则被推入大池。

如果没有使用磁带 I/O 从属，则会在 PGA 中分配用于磁带输出缓冲区的内存。设置 init.ora 参数 BACKUP_TAPE_IO_SLAVES 为 TRUE，可以使用磁带 I/O 从属，必要时还可以在服务器参数文件(SPFILE)中动态地设置该参数。BACKUP_TAPE_IO_SLAVES 参数设置为 TRUE 时，RMAN 会为每个信道创建一个从属进程来协助备份工作。为了协调这一功能，RMAN 会将内存分配推入 SGA。

如果配置了任一种 I/O 从属选项并且没有配置大池，则会从 SGA 的共享池区中分配内存。如果没有配置大池却又要使用 I/O 从属，我们建议最好创建一个大池，这个大池的大小基于为备份分配的信道总数(加上 1MB 用于开销)。第 16 章将介绍如何确定分配的信道数。如果已经具有用于共享服务器(多线程服务器)和 JDBC 连接池的大池，或者将 PARALLEL_AUTOMATIC_TUNING 参数的值设置为 TRUE，则可以根据 RMAN 内存缓冲区来增加大池的大小。

上面这些对 RMAN 内存体系结构的介绍未涉及根据 RMAN 备份来调整系统的内容。显而易见，运行 RMAN 时会发生资源占用。事实上，可以根据需要调整 RMAN 使用更多或更少的资源。第 16 章将详细讨论实现这个功能的方法。

有关内存使用的最后一个注意事项是：如果在磁带上做备份，就需要使用一个介质管理服务器(Media Management Server)产品。如果从与目标数据库相同的系统运行介质管理器(Media Manager)，则磁带子系统会需要额外的系统资源。调整备份时一定要考虑到这个因素。

Oracle SGA 中的大池

大池是 Oracle 内存空间的系统全局区(System Global Area，SGA)中的一个特定区域。使用 init.ora 文件或 SPFILE 文件中的 LARGE_POOL_SIZE 参数可以配置大池，这个参数值被指定为一个字节数。对于某些需要共享空间且涉及共享池中常见操作的内存活动可以利用大池。占用大池的区主要限于 RMAN 内存缓冲区(如果使用了 I/O 从属)和用于连接池(connection pooling)的共享服务器。大池有时用于 Java 连接，如果 PARALLEL_AUTOMATIC_TUNING 参数(在 10g 中已不使用该参数)被设置为 TRUE，大池还会包含并行查询从属(parallel query slave)。

实际上，我们不一定需要大池。如果没有大池，所有可能占用大池的区会直接使用共享池中的空间。这并不是没有解决办法，但是最好将 RMAN 缓冲区分到 PGA 中其自己的独立空间中。这样一来，SQL 和 PL/SQL 分析以及其他普通的共享池操作都不会受到 RMAN 备份的影响，反之亦然。此外还可以更方便、更直接地调整 RMAN 的 Oracle 内存空间。

2.9 恢复目录

到目前为止，我们已经讨论了两个最重要的 RMAN 组件：RMAN 客户端实用程序和内部数据库数据包。不过，还存在另一个涉及 RMAN 备份的重要组件(虽然它是完全可选的)：恢复目录。

恢复目录存储的是与 RMAN 备份有关的元数据。在某种意义上，恢复目录可以看作是保存 RMAN 备份和恢复所需的控制文件中的相关信息的副本。我们可以在 Oracle 数据库中的用户模式下创建恢复目录，这个恢复目录仅仅是一些数据包、表、索引和视图。RMAN 中的再同步命令会使得按目标数据库控制文件中的内容刷新这些表中的数据。当然，其区别在于恢复目录可以包含企业中所有数据库的信息，而控制文件只包含关于它自己的数据库的信息。

为了使用恢复目录，首先要从 RMAN 连接到目标数据库，随后还要在 RMAN 中生成连接到恢复目录的第二个 Net 连接，如下所示：

```
Rman>connect target /
rman>connect catalog rman/password@rcat
```

在连接恢复目录的连接字符串中，要输入该 RMAN 目录用户的用户名和口令。与连接目标数据库不同，连接到恢复目录的连接不是一个 sysdba 连接，因此不需要 sysdba 权限。

一旦连接了恢复目录，就可以手动再同步恢复目录，或者任何备份操作都可以隐式地再同步恢复目录。再同步是指用目标数据库控制文件中的内容刷新恢复目录中表内的信息。

一个恢复目录可以存储多个目标数据库的信息，这样可以集中管理许多不同数据库的备份。通过 SQL*Plus 可以查询恢复目录的视图，由此确定在恢复目录中注册的每个目标数据库的备份数、备份大小和备份范围。

图 2-4 介绍了采用恢复目录时的网络拓扑结构。在恢复目录中存在两个数据包：DBMS_RCVMAN 数据包和 DBMS_RCVCAT 数据包。DBMS_RCVMAN 数据包与 SYS 模式中的 DBMS_RCVMAN 数据包形式是一样的，因此 RMAN 实用程序既可以使用恢复目录，也可以使用目标数据库控制文件来提供备份和恢复的有关信息，并且不必担心其实现方式的不同。

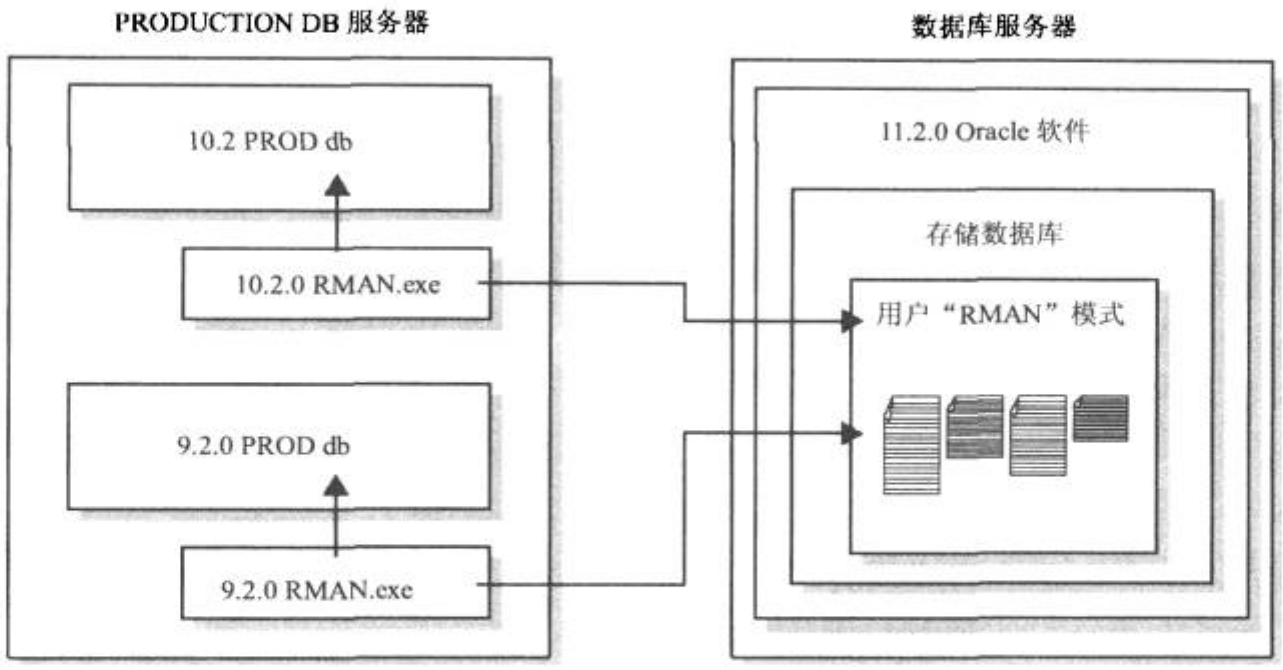


图 2-4　连接一个恢复目录

恢复目录中存在的数据包 DBMS_RCVMAN 会给包含该恢复目录的数据库带来混乱，这样的数据库通常称为目录数据库(catalog database)。目录数据库也可能是潜在的目标数据库，它也含有 SYS 模式中的 DBMS_RCVMAN 数据包，因此如果选择目录数据库中的 DBA_OBJECTS，就会存在模式不同但是名称相同的两个数据包。这样并不会发生错误或问题。因为其中一个数据包在数据库创建时由 catproc.sql 构建(在 SYS 模式下)，另一个数据包则是在创建恢复目录时构建(在常规的用户模式下)。

恢复目录中的第二个数据包是 DBMS_RCVCAT 数据包，这个数据包只在 RMAN 操作期间执行特定的恢复目录操作。实质上，可以将这个数据包看作恢复目录的 DBMS_BACKUP_RESTORE 数据包实现，区别仅在于 DBMS_BACKUP_RESTORE 数据包在目标数据库控制文件中写入备份完成信息，而 DBMS_RCVCAT 数据包则将这些信息写入恢复目录。

另外，包含恢复目录中的信息的基表(base table)并不重要，因为我们不想手动修改这些基表。为了保护恢复目录，Oracle 创建了一系列前缀为 RC_的视图，使用这些视图可以从恢复目录中析取信息。手动执行任何针对目录对象的 DML 命令是非常危险的，我们并不建议执行这样的操作。第 10 章简要介绍了 RC_*视图及其提供的信息。需要注意的是，这些视图是数据库控制文件中相应 v$视图的不同实现。

2.10　辅助数据库

辅助数据库(auxiliary database)是指在表空间时间点恢复(Tablespace Point In Time Recovery，TSPITR)、复制操作(克隆数据库)或使用 RMAN 备份创建备用数据库时，为了还原目标数据库的文件而临时生成的实例。执行上述任何一种任务时，我们都要在 RMAN 中同时连接目标数据库和辅助数据库，这样就可以利用目标数据库控制文件中的备份信息在辅助数据库位置调整这些备份的还原。下面所示的是连接到目标数据库(本地)

和辅助数据库(使用 Oracle Net 连接)的连接：

```
rman>connect target /
rman>connect auxiliary sys/pwd@aux1
```

执行这些命令后，RMAN 会同时生成到每个数据库的连接，并要求访问目标数据库与辅助数据库中的 SYS.DBMS_BACKUP_RESTORE 数据包和 SYS.DBMS_RCVMAN 数据包。这样，RMAN 要求辅助数据库具有与目标数据库相同的 sysdba 权限。由于 RMAN 必须生成到这两个数据库的 sysdba 连接，而且不可能本地连接两个不同的数据库，因此必须至少配置一个数据库，创建一个口令文件，并且生成到这个数据库的 Oracle Net 连接。

第 19 章将详细讨论辅助数据库的设置。图 2-5 介绍了使用辅助数据库时的 RMAN 配置网络拓扑结构。在 Oracle 8*i* 中的辅助数据库上执行任何动作时都需要恢复目录，因此图 2-5 显示了包含一个恢复目录的拓扑结构。

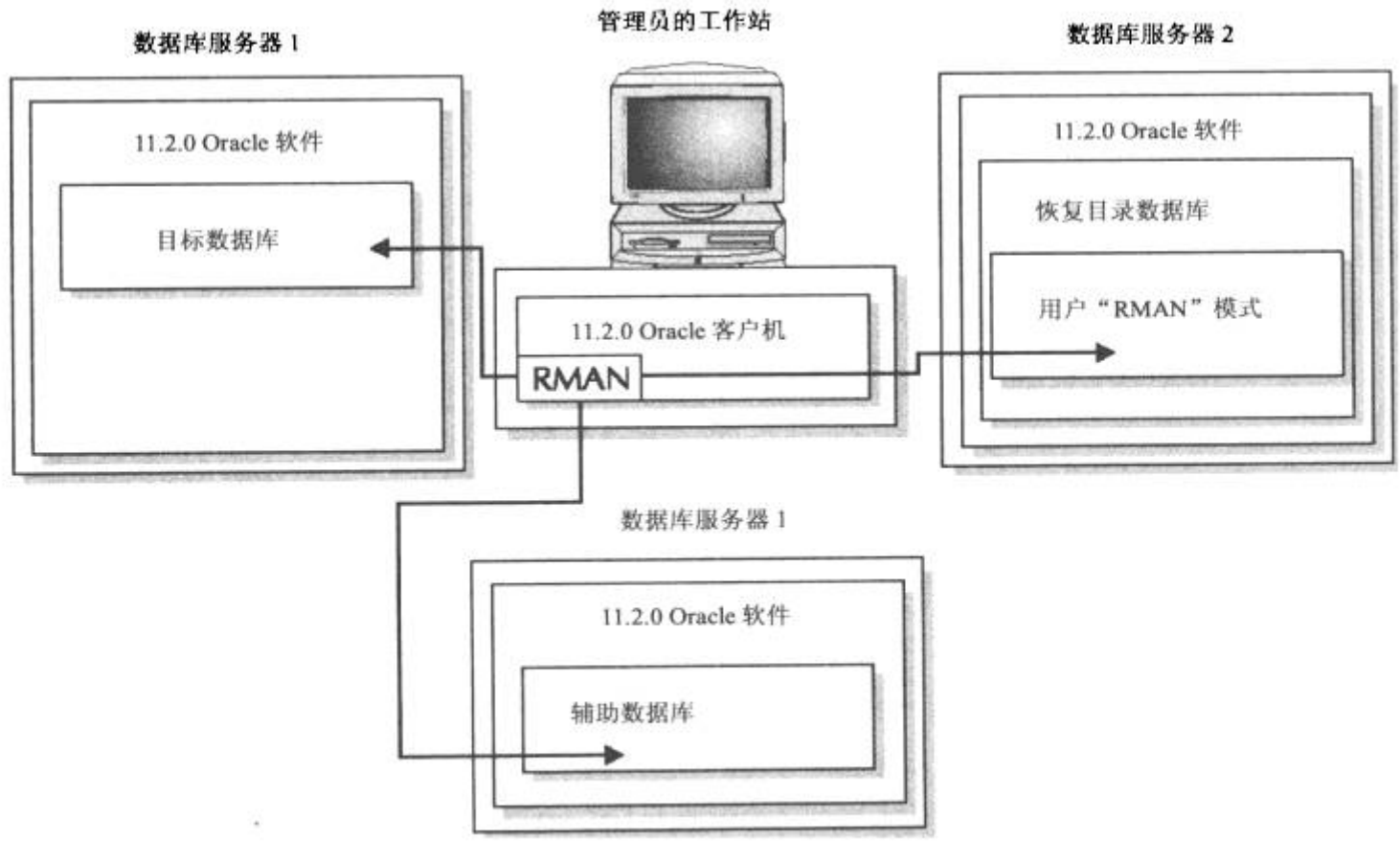


图 2-5 包含辅助数据库的网络拓扑结构

2.11 版本兼容性

鉴于必须使用很多种不同的组件，因此在使用 RMAN 时必须注意数据库版本的限制。版本兼容性问题分为 5 部分，每一部分都有一个版本号：

- RMAN 可执行文件版本(客户端实用程序)
- 目标数据库
- 恢复目录模式
- 恢复目录数据库

- 辅助数据库(用于复制、TSPITR 和备用数据库的创建)

当然，解决兼容性问题的最佳方法是保证这些组件都使用最新的 11.2 版本。如果使用相同版本的这些组件，一切都会井然有序。

当然，在所有数据库都是相同版本的情况下，所有事情都会按部就班，并且因为不会产生故障而不需要进行备份。但是，在实际生活中，还是需要了解关于 RMAN 版本兼容性的一些问题。

2.11.1　目标数据库与 RMAN 可执行文件

首先，要遵守的第一个通用规则是尽量保证目标数据库与 RMAN 可执行文件使用的是相同的版本。如果在目标数据库环境下运行 RMAN，做到这一点是很容易的。如果从一个集中的客户端接口运行所有的 RMAN 工作，这将非常棘手，这意味着客户端系统需要安装一个 ORACLE_HOME 客户端软件，版本上要对应需要连接和备份的每个数据库的版本，该解决方案相当复杂。使用 Oracle Enterprise Manager Grid Control 软件可以解决这个问题，该软件支持集中的客户端接口，因此我们可以从单个控制台使用远程 RMAN 可执行文件，或者在多个 DBA 完成备份任务时从多个控制台使用远程 RMAN 可执行文件。第 11 章将介绍 Enterprise Manager 接口。

2.11.2　恢复目录数据库与恢复目录模式

实质上，需要考虑 3 种版本的兼容性：Oracle 9*i*、Oracle 10*g* 和 Oracle 11*g*。针对恢复目录数据库与恢复目录模式的兼容性问题，一种简单的解决方法是：如果在 Oracle 11.2 版本的数据库中创建了一个 Oracle 11.2 版本的恢复目录，那么这个恢复目录可以注册 Oracle 8.1.7.4 版本以上的所有数据库。如果无法实现这一点，则版本向前兼容，即一个 8.1.x 版本的恢复目录可以注册所有 Oracle 8 版本的数据库。但是，Oracle 8*i* 版本的恢复目录不能注册 Oracle 9.0.1 版本或 Oracle 9.2.0 版本的数据库。Oracle 10*x* 版本的数据库也不能注册到 Oracle 9*x* 版本的目录中，11*g* 版本的数据库也不能在低版本的目录中注册。因此，如果数据库不能在 Oracle 10*g* 恢复目录中注册，就需要在 NOCATALOG(非目录)模式下运行和使用这个数据库。不要考虑运行多版本的目录——不需要这样做。记住，当在您的系统环境中引进一个新版本的 RDBMS 时，同时需要获得该版本的 RMAN 目录。

2.11.3　辅助数据库

从兼容性角度来看，辅助数据库必须与要复制的目标数据库采用相同的版本。实际上，最好使复制的 ORACLE_HOME 主目录匹配为与目标数据库的 ORACLE_HOME 主目录相同的版本。第 17 章将详细讨论辅助数据库的使用方法。

2.12　RMAN 进程

到目前为止，我们已经讨论了使用 RMAN 来执行备份的不同组件。您可能会注意到，备份操作分为多个步骤。为了使您对备份操作有一定的了解，我们将执行一个典型的备

份操作，并且解释每个步骤发生的内在 RMAN 活动。同时，您应当将这些实际步骤与本章介绍过的各种内容联系起来。

下面将举例说明对 PROD 数据库的备份操作，备份将存储在磁盘上。本书的第 4 章到第 8 章将介绍如何设置和使用磁带备份用到的介质管理器。目标数据库 PROD 有 20 个数据文件，并在 ARCHIVELOG 模式中运行。在备份操作期间，PROD 数据库继续运行。下面是备份操作的命令：

```
C$>rman
rman>connect target /
rman>backup database;
```

上面就是备份操作的所有命令，接下来将详细讨论备份操作。

RMAN 生成连接到目标数据库(已经在我们的环境中设置)的 bequeath 连接，这就是说 RMAN 会检查 ORACLE_SID 变量中的实例名，并在该实例上产生一个服务器进程，然后作为 sysdba 用户登录。这样，我们被连接为内部数据库用户 SYS，RMAN 会立即产生一个用于执行备份操作的信道进程。在这个示例中，由于使用默认的设置，因此只会分配一个信道。我们没有使用 I/O 从属，因此信道进程会在 PGA 中分配内存。

随后，RMAN 通过编译一个对 SYS.DBMS_RCVMAN 的调用，从目标数据库控制文件中请求数据库结构信息，先是确定目标数据库的版本。RMAN 从控制文件中收集版本信息和控制文件本身的信息，控制文件本身的信息包括控制文件的类型、控制文件中的当前序列号以及控制文件的创建时间。

由于指定了数据库的完整备份，所以 RMAN 会请求数据库中每个数据文件的信息，并且判断是否存在离线的数据文件。这些数据文件信息中包括每个数据文件所在磁盘和工作方式的信息。由于使用了默认设置，所以只存在一个信道和一个备份集。这样，RMAN 会忽略所有的磁盘相似性信息，把注意力放在编译包含在备份集内的文件列表上。

编译了文件列表之后，RMAN 就准备开始执行备份进程。为了保证一致性，RMAN 必须构建快照控制文件。如果存在快照控制文件，则 RMAN 会将这个文件重写为新的快照控制文件。接下来，RMAN 创建对 DBMS_BACKUP_RESTORE 数据包的调用，该调用可以创建备份片。备份片在默认的文件位置中构建，UNIX 系统中的这个文件的默认位置是 ORACLE_HOME/dbs，而在 Windows 系统中，这个文件的默认位置是 ORACLE_HOME/database。RMAN 拥有文件列表，因此它可以为磁盘读取操作分配内存缓冲区。PROD 数据库拥有 20 个数据文件，这样 RMAN 会分配一些大小为 128KB 的输入缓冲区，由于每个文件有 4 个输入缓冲区，因此这些输入缓冲区总共需要使用 10MB 的内存。RMAN 将只分配 4 个输出缓冲区，每个输出缓冲区的大小为 1MB。这样，备份操作需要使用的总内存为 14MB。

分配了内存之后，RMAN 初始化备份片。备份片会得到一个唯一的默认名。随后，RMAN 开始备份。在数据库版本 9.2、10.1 和 10.2 中，RMAN 以 50MB 为增量分配磁盘空间：在磁盘上分配 50MB 的空间并且填充输出缓冲区；当输出缓冲区已填满时，则获取另一个 50MB 空间，直到将最后一个数据块转储到备份片。当备份完成时，释放最后一个 50MB 数据块中的剩余空间。有必要指出的是，RMAN 不再检查是否存在完成所有

备份操作的足够空间。这是由于空值压缩和新的 10g 空白压缩可极大地减少备份中的数据文件大小。相反，RMAN 将运行它的备份直到空间用完为止，然后就会运行失败。

一旦初始化了备份片，信道进程就可以开始数据库备份进程。RMAN 会判断是否使用了服务器参数文件(SPFILE)，如果使用了服务器参数文件，则 RMAN 会自动将该文件备份为备份集的一部分。RMAN 还要在备份集中备份当前控制文件。只要备份了 SYSTEM 表空间，就会自动备份当前控制文件，但如果打开了控制文件自动备份(control file autobackup)功能，情况就会有所变化(参阅第 9 章)。

完成服务器参数文件和控制文件的备份后，就应当开始读取数据文件并将数据块推入内存。为了实现这一功能，信道进程在磁盘上执行预读(read-ahead)操作，并且将多个数据块同时推入内存。随后发生的是从输入缓冲区到输出缓冲区的内存对内存的写操作，在这个写操作期间，RMAN 会判断数据块是否初始化，或者数据块头信息是否仍然为零。如果数据块未被使用过，就不会发生到输出缓冲区的写操作，同时会丢弃这个数据块。如果数据块被使用过，则 RMAN 会在这个数据块上执行检验和(checksum)操作。如果数据块头和脚注不匹配，则 RMAN 会指出存在损坏的数据块并中止备份。如果数据块初始化并通过了检验和操作，这个数据块就会被写入输出缓冲区。

一旦输出缓冲区被填满，就会将输出缓冲区中的内容转储到备份文件位置。由于 RMAN 缓冲区填充来自所有数据文件的数据块，所以在转储文件中的数据块是杂乱无序的。转储文件只是一个存储段，只有 RMAN 能够在还原时将这些数据块还原为正确的位置。在备份片中写入数据块时，可以从 RMAN 影子(shadow)进程中得到备份的状态信息。RMAN 影子进程在目标数据库上检查远程进程调用(RPC)，并将信息传递给 V$SESSION_LONGOPS 视图以供用户查看。基于在备份操作开始时收集的信息，RMAN 会估计每个信道进程完成的百分比，这可以在下面的 V$SESSION_LONGOPS 视图中查看：

```
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
       ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"
FROM V$SESSION_LONGOPS
WHERE OPNAME LIKE 'RMAN%'
AND OPNAME NOT LIKE '%aggregate%'
AND TOTALWORK != 0
AND SOFAR <> TOTALWORK
/
       SID    SERIAL#    CONTEXT      SOFAR TOTALWORK  %_COMPLETE
---------- ---------- ---------- ---------- ---------- ----------
        17        167          1       4784     116328       4.11
```

在备份进程期间，我们还可以重新执行这样的查询，并可以得到剩余工作的最新信息，如下所示：

```
       SID    SERIAL#    CONTEXT      SOFAR  TOTALWORK   %_COMPLETE
---------- ---------- ---------- ---------- ---------- -----------
        17        167          1      96999     116328       83.38
```

一旦一个数据文件的所有数据块都被读入输入缓冲区并且确定了状态，RMAN 就会通过将这个数据文件头写入备份片来结束该文件的备份操作。在所有数据文件头都被写入备份片之后，RMAN 会生成最后一个对 SYS.DBMS_BACKUP_RESTORE 数据包的调用，该调用在控制文件中写入备份信息，这些信息包括备份片名、启动备份操作时的检查点的 SCN 和完成备份操作的时间。

上面介绍的就是备份操作的所有进程。显然，如果选择更多的备份选项(如使用多个信道、使用 FILESPERSET 参数及备份到磁带等)，备份操作将会更加复杂。不过，每种配置用到的基本进程都与前面所介绍的相同。在学习和测试 RMAN 时，如果要进一步了解备份期间 RMAN 内部的工作流程，则可以打开备份操作的调试选项，从而得到一个完整的进程列表：

```
Rman target / debug trace=/u02/oradata/trace/rmanbkup.out
```

需要注意的是，这个输出列表非常详细，并可能妨碍备份操作的性能。最好只在测试实例时使用这个列表来进行调试，或者只在检查产品备份问题时在 Oracle Support Services 的指导下使用这个列表来进行调试。

2.13 闪回恢复区

闪回恢复区(FRA)不是使用 RMAN 所需的组件，但它应该成为 RMAN 所需的组件。作为 10g 版本的新增内容，FRA 是磁盘上所设定的用于存储所有 Oracle 恢复文件的特定位置。恢复文件是指进行介质恢复操作时所需的所有文件：完全的数据文件备份、增量备份、数据文件副本、备份控制文件以及归档的日志。FRA 也存储联机重做日志文件、块更改跟踪文件以及当前控制文件的镜像副本。如果进行设置，FRA 中也存储使用闪回数据库选项的闪回日志(在第 13 章中将讨论闪回技术)。

FRA 的潜在作用是简化备份和恢复工作的管理，方法是将必需的文件合并到一个位置，从而使 Oracle 和 RMAN 可以进行细节管理，而 DBA 就可以转为处理其他重要的工作。这种简化是基于关注可用性的牢固备份策略的一些潜在原理：

- 在数据库可以进行本地访问的磁盘上应该保存重要数据文件的至少一个副本(如果没有保存整个数据库的副本的话)。
- 根据本地磁盘的存储压力，应该将一段时间以前的备份转移到磁带上。
- 根据业务规则，长期的备份管理应该是自动进行的。

您设置的 FRA 可以是普通磁盘卷上的目录或自动存储管理(ASM)磁盘组。FRA 通过两个初始参数确定：DB_RECOVERY_FILE_DEST 和 DB_RECOVERY_FILE_DEST_SIZE。第一个参数确定其位置，第二个参数则确定其大小。可以在 init.ora 文件(如果仍然使用该文件)中设置这些参数，或者在 SPFILE 中通过执行 alter system set 命令设置这些参数。

通过配置 FRA，我们就不需要为归档的日志设置任何其他的 LOG_ARCHIVE_DEST_n 参数；对于 FRA，Oracle 将默认值 LOG_ARCHIVE_DEST_10 分配给 FRA。也应该注意，使用 FRA 时，不可以使用 LOG_ARCHIVE_DEST 或 LOG_ARCHIVE_DUPLEX_ DEST；

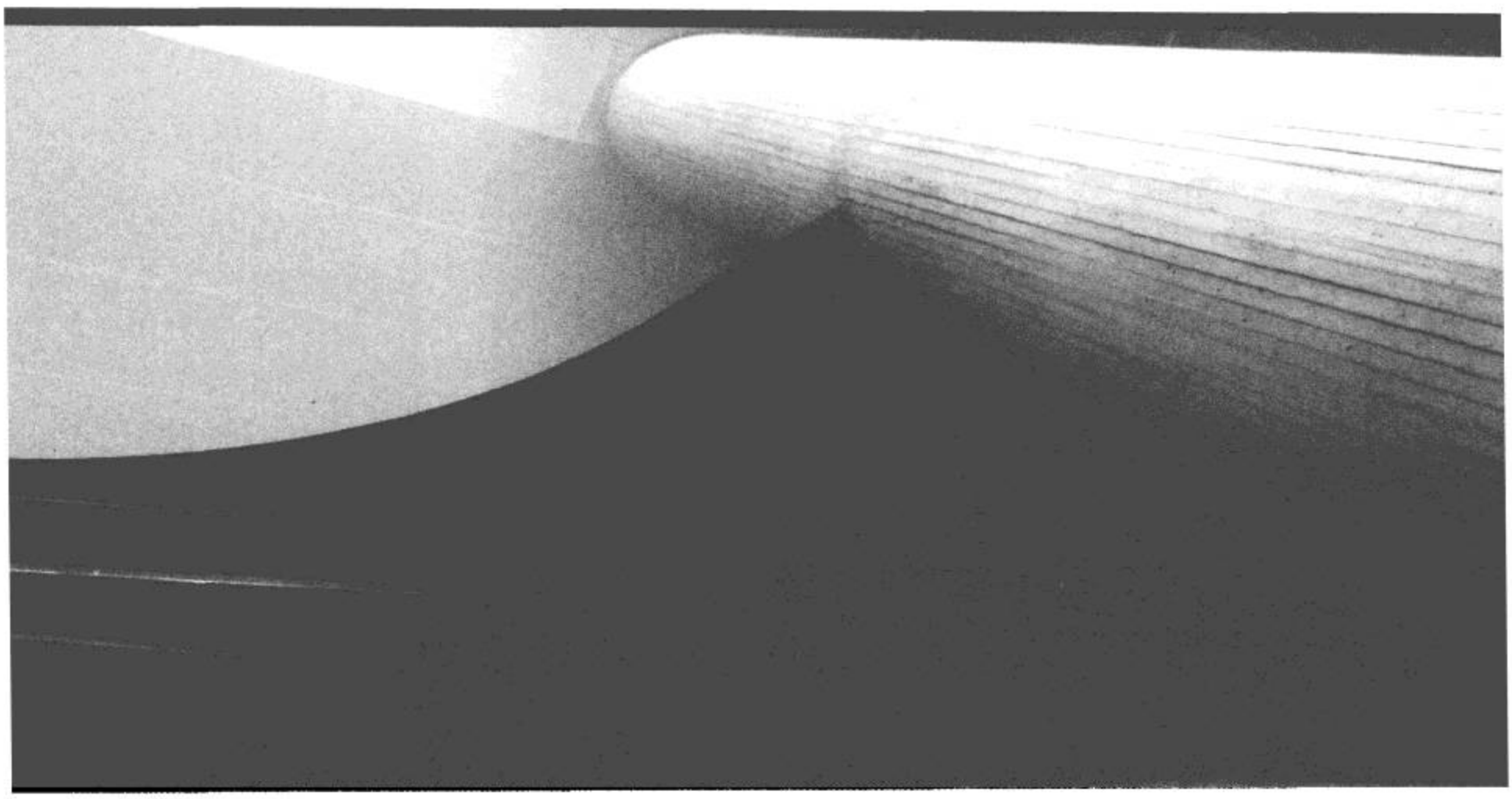
不过，您应该已经很久没有使用过这些过时的参数了。

FRA 在内部管理恢复文件首先是基于数据库名，然后基于文件类型，接下来是文件的生成日期。文件本身则根据 Oracle 管理文件(Oracle Managed Files，OMF)格式命名。这样，这些文件就很难解密(除了归档的日志，它仍然保留使用 LOG_ARCHIVE_FORMAT 参数提供的结构)。也有一些重要的内部目录结构因文件管理的需要而存在。然而，FRA 的作用在于不需要花费过多的时间来操心文件本身。这也就是说，应该关注内部的结构并且熟悉文件的位置。因为您早晚都会要用手动来查找特定文件。

多个数据库可以使用相同的 FRA。这可以提供极大的好处，特别是对于数据保护(Data Guard)配置。如果在相同系统上有较大的 ASM 磁盘组和多个数据库，也可以使用该方法。当需要复制数据库以进行测试时，使用该方法也会非常方便。需要注意的是，使用相同 FRA 的所有数据库必须具有不同的 DB_NAME 值，或者对于值 DB_UNIQUE_NAME 具有不同的名称。

2.14　小结

本章介绍了 RMAN 执行 Oracle Database 11g 数据库备份操作的内在体系结构，涉及到 RMAN 可执行文件、目标数据库数据包和控制文件，并且详细讨论了进程体系结构和如何分配用于 RMAN 备份的内存，还介绍了 RMAN 恢复目录的用途，以及连接辅助数据库的方法。结束对这些不同组件的介绍之后，我们简要描述了一个典型的备份操作流程，并说明了在流程中使用的各种不同组件。



第 II 部分

设置原则和实践

第 3 章　RMAN 设置和配置

第 4 章　介质管理问题

第 5 章　Oracle 安全备份

第 6 章　使用 OSB 云模块将数据库备份到 Amazon Web 服务

第 7 章　使用 Oracle 的 VERITAS NetBackup 增强 RMAN

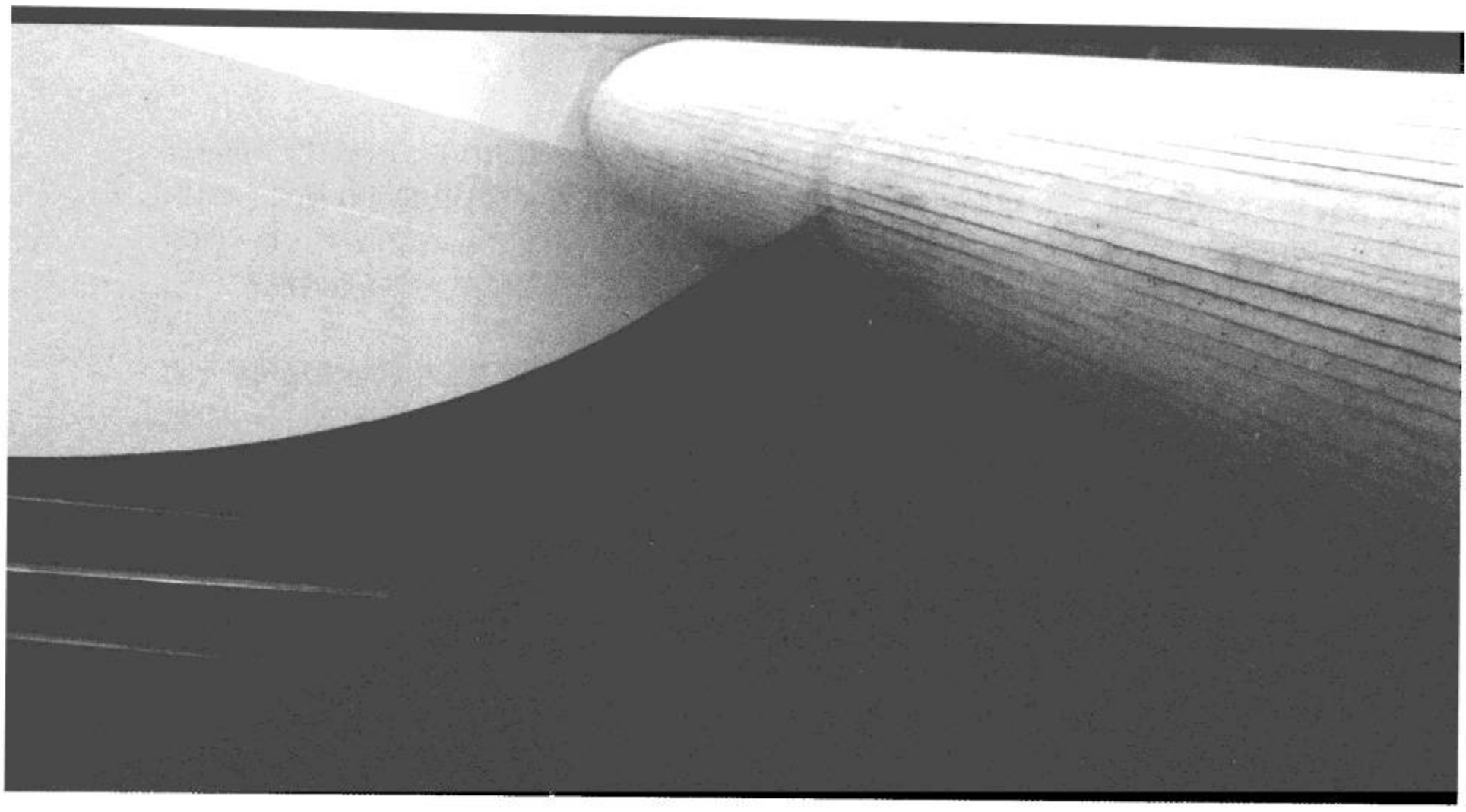
第 8 章　配置 Oracle 的 HP Data Protector

第 9 章　RMAN 与 Tivoli Storage Manager

第 10 章　恢复目录的使用

第 11 章　RMAN 备份

第 12 章　RMAN 还原与恢复



第3章

RMAN 设置和配置

在使用 RMAN 执行实际的备份和恢复操作之前，首先需要设置用于备份和恢复操作的 RMAN 与数据库。本章将介绍最初的 RMAN 设置要求和选项。首先，由于重做日志在 Oracle 数据库恢复中具有关键地位，因此我们将在第 1 章关于 Oracle 重做日志介绍的基础上，进一步深入讨论 Oracle 重做日志。在上述讨论的基础上，将学习为执行联机备份操作而将数据库置入 ARCHIVELOG 模式的情况，然后再介绍基本的 RMAN 接口，使您对 RMAN 有进一步的认识。接下来，将讨论如何为数据库备份操作对 RMAN 进行配置。最后，还要讨论 RMAN 恢复目录，包括使用恢复目录的原因以及配置恢复目录的方法。

3.1　配置数据库以在 ARCHIVELOG 模式中运行

在第 1 章，我们已经介绍了 ARCHIVELOG 模式和 NOARCHIVELOG 模式，了解 redo 对数据库的重要性，了解为何多数 DBA 将其数据库设置为 ARCHIVELOG 模式的原因。如果您认为自己的数据库可以设置为 NOARCHIVELOG 模式，则本节的多数讨论不适合应用于该种方式。如果您打算采用 ARCHIVELOG 模式，则需要做一些基本的配置工作。本章主要讨论这些配置工作。

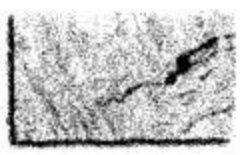
当数据库在 ARCHIVELOG 模式运行时，归档重做日志的复制位置可采用两种配置方法。实际上，也可以同时采用两种方式。第一种选择是配置 ARCHIVELOG 目标目录，第二种选择是配置 Oracle 闪回恢复区(Flash Recovery Area，FRA)。在后续内容中，首先将讨论这两种配置，然后讨论将数据库配置为在 ARCHIVELOG 模式中运行的实际操作方法。

3.1.1　ARCHIVELOG 目标目录

在配置 ARCHIVELOG 模式时，需要确定让 Oracle 在何处创建归档重做日志。选择使用归档日志目标目录可长期有效。要采用归档日志目标目录方式，需要在 Oracle 中设置一些特定参数来配置选项。首先，可使用 LOG_ARCHIVE_DEST_n(n 是 1~10 的一个范围值)参数定义最多 10 个不同的归档日志目标，这些目标可以是本地目录、网络目录(如 NT 文件夹)或网络连接存储(NAS)，如果使用 Oracle 的备用数据库/数据保护产品，目标还可能是一个定义的数据库服务器名。需要注意的是 LOG_ARCHIVE_DEST_n 参数没有默认定义。

如果使用 SPFILES，则可以使用 alter system 命令设置 LOG_ARCHIVE_DEST_n 参数，示例如下：

```
alter system set log_archive_dest_1 = 'location = c:\oracle\oraarc\beta1';
```



注意：
在该设置中，常见的错误是设置的 LOG_ARCHIVE_DEST 目录不存在，或者 Oracle 不能对该目录执行写操作。为确保正确设置该参数并且数据库处于 aRCHIVE 模式，可执行 alter system switch logfile 命令，以确保 ARCH 进程正确地执行写归档重做日志的工作。

每一个 LOG_ARCHIVE_DEST_n 位置都可以定义为强制或任选两种方式之一。在 Oracle Database 11g 中，LOG_ARCHIVE_DEST_n 位置默认定义为任选方式。如果定义为强制方式，则意味着归档重做日志必须写到该位置。当 ARCH 进程在对采用强制方式的位置写入发生错误时，数据库所有活动将被快速挂起(在循环通过所有联机重做日志后)。任选重做位置对数据库操作没有影响。

```
alter system set log_archive_dest_1 = 'location = c:\oracle\oraarc\beta1
    mandatory';
```

在 Oracle Database 11*g* 中，所有 LOG_ARCHIVE_DEST_n 参数的位置在默认情况下都是可选的(尽管位置总是按顺序排列的，因为 LOG_ARCHIVE_MIN_SUCCEED_DEST 最小的设置是 1)。参数 LOG_ARCHIVE_MIN_SUCCEED_DEST 指定，归档多少日志目标目录中必须包含联机重做日志的成功复制的副本才能认为是已成功归档。LOG_ARCHIVE_MIN_SUCCEED_DEST 参数默认的设置是 1，并且是该参数的最小设置。以下是该参数值设置为 2 的示例。

```
alter system set log_archive_min_succeed_dest=2;
```

除了配置 LOG_ARCHIVE_DEST 参数，可能还需要考虑配置与数据库归档重做日志、ARCH 进程和 LOG_ARCHIVE_OEST 一系列参数相关的如下参数。

- **LOG_ARCHIVE_STATE_n 参数**：为每个归档日志目标定义两种不同状态中的一种状态。如果设置为 ENABLE，ARCH 进程会认为与该状态关联的目标为一个有效的归档日志目标；如果设置为 DEFER，ARCH 进程就不会在相关的 LOG_ARCHIVE_DEST_n 位置归档日志。
- **LOG_ARCHIVE_FORMAT 参数**：在命名归档的重做日志时提供一个供 Oracle 使用的模板。Oracle 创建归档的重做日志时，按以下方式重命名这些归档的重做日志，即每个归档的重做日志都有唯一的名称。可以使用 LOG_ARCHIVE_FORMAT 参数来设置所需的默认命名标准。该参数对在 FRA 中创建的归档重做日志没有影响。
- **LOG_ARCHIVE_START 参数**：该参数在 Oracle Database 10*g* 和其后版本中不再使用。Oracle 现在会自动启动 ARCH 进程。
- **LOG_ARCHIVE_MAX_PROCESS 参数**：该参数定义在 Oracle 数据库启动时 ARCH 进程的数量。

注意：
如果您使用的是 Oracle Database 9*i* 或更早的数据库版本，在将数据库配置为 ARCHIVELOG 模式时，必须将 LOG_ARCHIVE_START 参数设置为 TRUE。在 10*g* 以后的版本中，不再使用该参数。

目前提到的这些参数在 *Oracle Database 11g Reference Manual* 中有定义(它是 Oracle 文档的一部分)，该指南介绍了更多的信息。

在接下来的示例中，有一个需要设置为 ARCHIVELOG 模式的数据库。我们将创建 3 个不同的归档日志目标目录，其中一个被指定为支持 Oracle 备用数据库的服务器名。我们还将实现这样的要求，即至少有两个目录必须被写入以便完成归档的重做日志的复制，并且备用数据库必须是其中一个目录。下面是与 ARCHIVELOG 模式操作相关的数据库参数文件的多个参数的使用示例。

```
log_archive_dest_1='location=d:\oracle\oraarc\robt mandatory'
log_archive_dest_2='location=z:\oracle\oraarc\robt optional'
log_archive_dest_3='service=recover1 mandatory'
log_archive_min_succeed_dest=2
log_archive_format="robt_%s_%t.arc"
```

在这个示例中，第一个归档日志目标位置为 d:\oracle\oraarc\robt；第二个归档日志目标位置为 Z:驱动器，由于这是一个网络设备(它不一定可靠)，所以设置其为可选的位置；第三个归档日志目标位置为一个名为 recover1 的 Oracle Net 服务器(可能是备用数据库)，在生成归档的重做日志时，该目标会使得 Oracle 通过 Oracle Net 发送这些归档的重做日志。

在上面的示例中，我们通过使用 LOG_ARCHIVE_MIN_SUCCEED_DEST 参数指定归档的重做日志必须成功地复制到至少两个不同的位置上；LOG_ARCHIVE_FORMAT 参数定义了归档的重做日志的格式。

3.1.2 闪回恢复区

闪回恢复区(FRA)允许集中存储所有与恢复相关的文件。FRA 可以本地使用附加的存储、Oracle 的群集文件系统(Oracle Cluster File System，OCFS)或 Oracle Database 10g 中新的自动存储管理(ASM)特性。表 3-1 列出了可以包含在 FRA 中的备份文件类型。

表 3-1 闪回恢复区中的文件类型

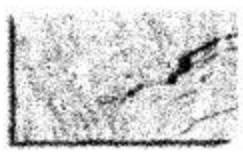
文件类型	说明
归档的重做日志	归档的重做日志将存储在 FRA 中
控制文件	当创建数据库时，在 FRA 中创建控制文件的一个副本
控制文件自动备份	RMAN 控制文件自动备份的默认位置是 FRA(如果定义了 FRA)
闪回日志	闪回日志(本章后面将讨论)将存储在 FRA 中(如果定义了 FRA)
重做日志	每个重做日志组成员的一个副本可以存储在 FRA 中
RMAN 数据文件副本	RMAN 数据文件副本的默认位置将是 FRA(如果定义了 FRA)
RMAN 备份和其他相关文件	RMAN 常规文件(备份集片等)的默认位置将是 FRA(如果定义了 FRA)

FRA 帮助管理全部的磁盘空间分配，并且为所有与恢复相关的文件提供集中的存储区域。FRA 也提供更快速的备份和恢复操作。

FRA 中文件的保留与否由 RMAN 保留策略决定。通过执行 RMAN configure retention policy 命令设置该策略。本章将详细讨论该命令和 RMAN 保存。如果文件没有关联的保留策略，或者是永久的文件，则该文件永远不会被删除。如果文件在 RMAN 保留策略下没有过时，则不会被删除。最后，如果归档的日志过时，则可以删除它们。

FRA 在由参数 DB_RECOVERY_FILE_DEST 定义的特定位置中创建。该位置可以是文件系统或 ASM 卷。DB_RECOVERY_FILE_DEST_SIZE 参数则用来定义 FRA 的最大尺

寸。这是 Oracle 控制的文件空间限制，而与文件系统自身中的全部可用空间无关。Oracle 监控 FRA 中的可用空间，一旦 FRA 中的可用空间数量减少到不安全的程度，Oracle 就会在警报日志中生成警告。在 Oracle Database 11g 中，当可收回的空间低于 DB_RECOVERY_FILE_DEST_SIZE 参数定义值的 15%时，将会生成警报，当低于 3%时将产生危险性警报。这些警报将会出现在 OEM 中的警报日志中，也可以通过 DBA_OUTSTANDING_ALERTS 表获知。如果希望知道您的闪回恢复区空间使用情况时，可以通过几个视图获取该类信息。下面将详细讨论这些视图。



注意：
如果闪回恢复区是唯一的归档日志目标，那么 FRA 中的空间被用完时就会非常麻烦，因为这可能造成数据库最终异常终止。如果 FRA 将是唯一的归档日志目标，则应该仔细地监控空间可用性。

为 FRA 分配多少空间合适是一个难题。一般可以通过监视空间使用情况，按照 FRA 需要对空间大小进行调整。在数据库已经构建完成时，可以检查 DBA_HIST_LOG 视图获得归档日志空间使用的情况。该视图的数据来源于 Oracle 的 AWR 基础架构。通过该视图可以获知归档重做日志的平均容量以及日志切换时间。下面给出查询 DBA_HIST_LOG 视图的示例：

```
SQL> alter session set nls_date_format='mm/dd/yyyy hh24:mi:ss';
Session altered.

SQL> select sequence#, first_time Log_started
  2 ,lead(first_time, 1,NULL) over (order by first_time) Log_ended
  3 from (select distinct sequence#, first_time
  4 from dba_hist_log
  5 where archived='YES'
  6 and sequence#!=0
  7 order by first_time)
  8 order by sequence#;
SEQUENCE# LOG_STARTED LOG_ENDED
--------- ------------------- -------------------
       82 05/18/2009 22:02:56 05/19/2009 12:52:06
       83 05/19/2009 12:52:06 05/19/2009 22:01:24
       84 05/19/2009 22:01:24 05/20/2009 15:25:39
       85 05/20/2009 15:25:39 05/21/2009 10:00:58
       86 05/21/2009 10:00:58 05/21/2009 17:02:00
       87 05/21/2009 17:02:00 05/21/2009 22:01:28
       88 05/21/2009 22:01:28

7 rows selected.
```

许多大型系统往往运行多个 Oracle 数据库。每个数据库都会使用归档日志存储空间。通常所有数据库产生的归档重做日志将会存储在同一个文件系统中。有时某个数据库可能会将存储设备上的所有空间都消耗殆尽。此时所有使用该存储设备的数据库都会受到

影响，因为它们已没有用于建立其归档重做日志的空间。使用 FRA 的好处在于，它能够为每个数据库分配特定的可用空间配额。由此，若使用 FRA，将会消除由于单个数据库消耗所有归档日志空间而影响其他所有数据库的风险。

如果发现 FRA 中无可用的空间，则有一些不同的方法可应对该问题：

(1) 如果问题是因 DB_RECOVERY_FILE_DEST_SIZE 参数导致的空间分配不足，并且存在足够的物理磁盘空间来增加分配给 FRA 的空间，则增加该参数值的大小。这将立刻为 FRA 增加额外的空间。当然，不能将该参数的值设置为大于文件系统中的可用物理空间数量。

(2) 如果需要更多的物理空间，则分配额外的物理空间给文件系统，然后增加 DB_RECOVERY_FILE_DEST_SIZE 参数值的大小。

(3) 如果没有额外的空间，可以将 FRA 移动到具有较多可用空间的另一个文件系统。

(4) 也可以在 FRA 中留出空间，方法是使用 RMAN backup recovery area 命令将 FRA 的内容移动到另一个位置。将在讨论执行 RMAN 备份时介绍 backup recovery area 命令和其限制。

(5) 最后一个方法是，物理删除 FRA 中较早的备份集片和/或 FRA 的归档的重做日志，然后使用 RMAN crosscheck 命令来让数据库认可已经删除了这些文件。

注意：

如果对是否删除 FRA 中的物理文件犹豫不决，则应该相信您的直觉。本质上，这意味着您的保留策略不正确或没有分配足够的空间来支持为数据库建立的保留策略。同样，删除文件会潜在地损害数据库的可恢复性，因此在删除文件时需要特别谨慎。

设置闪回恢复区

为了设置 FRA，需要配置如表 3-2 所示的参数：

表 3-2 设置 FRA 所需配置的参数

参 数	示 例	目 标
DB_RECOVERY_FILE_DEST_SIZE	Alter system set db_recovery_file_dest_size=20G scope=both;	设置 FRA 的已分配大小(以字节为单位)，并且必须定义具体的大小以启用 FRA。这就允许对分配给 FRA 的磁盘空间大小进行控制。不应该将这个值设置为大于全部可用磁盘空间数量的大小
DB_RECOVERY_FILE_DEST	Alter system set db_recovery_file_dest='/u01/oracle/flash_recovery' scope=both;	指定 FRA 的位置。这可能是文件系统、ASM 磁盘位置或 OMF 位置

注意，必须在指定 DB_RECOVERY_FILE_DEST 参数之前指定 DB_RECOVERY_

FILE_DEST_SIZE 参数。如果没有这样做，将会产生 ORA-32001 错误消息。类似的情况是，必须在复位 DB_RECOVERY_FILE_DEST_SIZE 参数之前禁用 DB_RECOVERY_FILE_DEST 参数。保留 DB_RECOVERY_FILE_DEST 参数为空将可以禁用 FRA。下面是通过复位 DB_RECOVERY_FILE_DEST 参数禁用 FRA 的示例。

```
alter system set db_recovery_file_dest=' ' scope=both;
```

Oracle 允许将归档重做日志归档到 FRA 及通过使用 LOG_ARCHIVE_DEST_n 参数定义的一个或多个附加位置，允许同时采用两种方法。当您配置了备份数据库并且希望利用 FRA 的优点时，您可能希望采取该种方法。

为了配置FRA和归档日志目标目录，需要设置标准的FRA参数DB_RECOVERY_FILE_DEST，以定义 FRA 的位置。还需要定义各种所需的 LOG_ARCHIVE_DEST_n 参数。默认情况下，当定义了 LOG_ARCHIVE_DEST_n 参数后，归档时将会使用该位置而不是使用 FRA。在此情况下，若在设置 LOG_ARCHIVE_DEST_n 参数时要使用 FRA，就需要为 FRA 定义一个额外的 LOG_ARCHIVE_DEST_n 参数。通常使用 LOG_ARCHIVE_DEST_10，使用 Oracle 支持的参数 USE_DB_RECOVERY_FILE_DEST。以此定义该位置是 FRA。如下例所示，Oracle 被配置为使用 FRA 和普通的归档日志目标目录：

```
alter system set log_archive_dest_10='LOCATION=USE_DB_RECOVERY_FILE_
 DEST';
alter system set log_archive_dest_1='location=c:\oracle\oraarc\beta1
 mandatory';
```

在该示例中，ARCH 进程将在 LOG_ARCHIVE_DEST_1 参数与 LOG_ARCHIVE_DEST_10 参数中建立归档日志，后者空间属于 FRA。

闪回恢复区视图

可以使用一些视图来帮助管理 FRA。这些视图包括：

- DBA_OUTSTANDING_ALERTS
- V$RECOVERY_FILE_DEST
- V$FLASH_RECOVERY_AREA_USAGE

同样，可以使用其他一些视图中的列来帮助管理 FRA。接下来详细介绍这些视图和列。

DBA_OUTSTANDING_ALERTS 视图

从 FRA 中添加或删除文件时，这些事件被记录到数据库警报日志中。可以检查新的 DBA 视图 DBA_OUTSTANDING_ALERTS 来了解关于 FRA 一些显著问题的信息。注意，空间相关问题产生的时间和警告出现在 DBA_OUTSTANDING_ALERTS 视图中的时间存在一定程度的延迟。

下面是 FRA 使用完空间并且提交警报给 DBA_OUTSTANDING_ALERTS 视图的示例。此时，您可能需要快速处理这种情况，否则数据库就可能完全停止。在这种情况下，可以使用 alter system 命令来增加分配给 FRA 的空间数量。

```
SQL> select reason from dba_outstanding_alerts;
REASON
----------------------------------------------------------------
db_recovery_file_dest_size of 524288000 bytes is 100.00% used
and has 0 remaining bytes available.

SQL> alter system set db_recovery_file_dest_size=800m;
```

V$RECOVERY_FILE_DEST 视图

V$RECOVERY_FILE_DEST 视图提供在数据库中定义的 FRA 的概述。该视图提供配置的 FRA 的大小、所使用的空间数量、可以回收多少空间，以及 FRA 中的文件数量。在下面的示例中，可以看到已经记录 FRA 的空间增加到 800MB(SPACE_LIMIT)。然而，仍然使用了过多的空间(SPACE_USED)并且 FRA 已满。

```
SQL> select * from v$recovery_file_dest;
NAME
-------------------------------------------------------------------------
   SPACE_LIMI  TSPACE_USED           SPACE_RECLAIMABLE NUMBER_OF_FILES
------------- ------------------- ----------------- -------------
c:\oracle\product\10.2.0\flash_recovery_area
   838,860,800       1,057,116,672       338,081,280            11
```

关于 Oracle 的一个较好的方面是，它会尽可能好地管理 FRA 空间，如果有可回收的空间，它就会根据需要释放这些空间。注意，在前面的查询中，Oracle 指示已经用完 FRA 空间。您是否注意到了 SPACE_RECLAIMABLE 列？该列指示有可回收的空间。该空间由归档的重做日志或备份集片占用，根据已经选择的保存标准(本章后面将讨论保存标准以及如何设置该标准)，已经不再需要这些空间。当 Oracle 需要 FRA 中的空间时(例如强制实施日志切换)，它将删除可回收的任何文件并释放空间。在下面的查询中，可以看到已经执行了这项操作。在运行前面的查询(该查询表明 FRA 空间被用完)后，强制实施一个日志切换。这会使 Oracle 回收 FRA 中的空间以重新使用，然后就能够写出归档的重做日志。可以查询 V$RECOVERY_FILE_DEST 视图并看到已经实际执行了这些操作：

```
SQL> alter system switch logfile;
System altered.
SQL> select * from v$recovery_file_dest;
NAME
-------------------------------------------------------------------------
   SPACE_LIMIT SPACE_USED           SPACE_RECLAIMABLE NUMBER_OF_FILES
------------- ------------------- ----------------- -------------
c:\oracle\product\10.2.0\flash_recovery_area
   838,860,800         719,412,736            64,000             7
```

V$FLASH_RECOVERY_AREA_USAGE 视图

V$FLASH_RECOVERY_AREA_USAGE 视图提供了关于占用 FRA 空间的文件类型

的详细信息。该视图对文件类型分组，然后提供每个文件类型所使用的空间百分比、来自于该组的总 FRA 可回收空间百分比，以及来自于该组的 FRA 中的文件数量。下面是 V$FLASH_RECOVERY_AREA_USAGE 视图的查询：

```
SQL> SELECT * FROM V$FLASH_RECOVERY_AREA_USAGE;

FILE_TYPE    PERCENT_SPACE_USED PERCENT_SPACE_RECLAIMABLE NUMBER_OF_FILES
------------ ------------------ ------------------------- ---------------
CONTROLFILE                   0                         0               0
ONLINELOG                     0                         0               0
ARCHIVELOG                17.14                     17.09               7
BACKUPPIECE               108.88                     23.22               4
IMAGECOPY                      0                         0               0
FLASHBACKLOG                   0                         0               0
```

在这个示例中，可以注意到：

- 超出了已定义的空间分配(所有行的 PERCENT_SPACE_USED 超过 100%)。造成这种情况的原因可能是 FRA 容量在近期被修改过，Oracle 还没有将空间重新回收(如果该工作执行，则总的使用空间将低于 100%)。
- 备份集片占用了大多数的空间，并且该空间的 23.22%是可回收的。
- 归档的重做日志只使用分配给 FRA 的空间的 17%，即使删除所有的归档重做日志，也不会释放出足够的空间来使 FRA 小于分配给它的空间数量。

包含 FRA 列的其他视图

可以在许多 Oracle Database 的 V$视图中找到列 IS_RECOVERY_DEST_FILE，例如 V$CONTROLFILE 、 V$LOGFILE 、 V$ARCHIVED_LOG 、 V$DATAFILE_COPY 、 V$DATAFILE 和 V$BACKUP_PIECE。该列是一个布尔值，用来指示文件是否位于 FRA 中。

另一个列 BYTES 可以在 V$BACKUP_PIECE 和 RC_BACKUP_PIECE(RMAN 恢复目录视图)视图中找到。该列以字节为单位指示备份集片的大小。

注意：
手动删除 FRA 中固定的文件会产生无法预料的结果。Oracle 不会立刻检测出这些文件是否已删除，因此不会回收相应的空间。如果最终手动删除文件(或者释放磁盘)，则使用 RMAN crosscheck 命令结合 delete 命令来使 Oracle 更新 FRA 上的当前控制文件信息。Oracle 专家建议尽可能不要手动删除由 Oracle 管理的文件。

其他闪回恢复区特性

在 Oracle 数据库中，如果没有设置 OMF 相关的 DB_CREATE_ONLINE_LOG_DEST_n 参数，则 alter database add logfile 和 alter database add standby logfile 命令创建 FRA 中的联机重做日志成员。alter database drop logfile 和 alter database rename file 命令也支持 FRA 中的文件。使用 OMF 相关的特性带来的好处是 Oracle 将会为您管理物理文件，当

删除联机重做日志组，而该组包含的物理文件是由 Oracle 基于 DB_CREATE_ ONLINE_ LOG_DEST_n 参数创建时，Oracle 将为您删除这些物理文件。

在数据库创建期间，Oracle 可以使用 FRA 存储数据库控制文件和联机重做日志。如果定义了 OMF 相关的参数 DB_CREATE_ONLINE_LOG_DEST_n，则将在这些位置创建控制文件和重做日志，但不会在 FRA 中创建，即使定义了 FRA。如果没有定义 DB_CREATE_ONLINE_LOG_DEST_n 参数，但定义了 CREATE_FILE_DEST 参数，则在由 CREATE_FILE_DEST 参数定义的位置中创建控制文件和联机重做日志。如果也定义了 DB_RECOVERY_FILE_DEST 参数，则也在相应位置创建控制文件和联机重做日志的副本。其结果是多路复用的联机重做日志。最后，如果只定义了 DB_RECOVERY_FILE_ DEST，则在该位置创建控制文件。如果没有定义任何这些参数，则在 OS 特定的默认位置创建控制文件和联机重做日志。

FRA 还有另外一种使用特性，该特性与 Flashback Database 有关。在第 15 章将对 Flashback Database 进行详细地讨论。

FRA 和 ASM

RMAN 支持使用自动存储管理(ASM)存储 RMAN 备份。ASM 是磁盘管理工具，使用该工具，DBA 就不再需要管理与给定数据库关联的物理文件。ASM 在一定程度上类似于可用于 UNIX 系统中的逻辑卷组。ASM 使用 ASM 磁盘组，该磁盘组是存储的逻辑单元。物理磁盘被分配给 ASM 磁盘组，构成 ASM 磁盘组的整体存储功能。ASM 磁盘组可以存在于前面分配的文件系统上，或者存储在原始磁盘上。结合 OCFS，群集服务器可以共享 RAC 配置中的 ASM 磁盘。配置 ASM 并定义各种磁盘组后，就可以分配数据文件、控制文件、联机重做日志以及各种 RMAN 备份文件到 ASM 磁盘组。

ASM 提供了大量特性，包括负载平衡、数据冗余，并且很容易添加和删除 ASM 磁盘组中的新磁盘。讨论 ASM 的常规配置超出了本书的范围。然而，需要注意的是，RMAN 支持 ASM 磁盘组，因此可以在必要时使用它们。在本书中并不建议使用 ASM。大多数非 RAC 站点使用 ASM 实现并没有多少价值。然而，如果是 RAC 站点，就会希望考虑使用 ASM 结合 OCFS 作为其他群集选项的替代方法，这取决于当前的平台。

如果正在使用 ASM，则可以配置 FRA，从而在 ASM 文件系统中创建 FRA，如同下面的示例所示：

```
alter system set db_recovery_file_dest='+ASMV01';
```

在该示例中，Oracle 将 ASM 磁盘卷 ASMV01 用作 FRA。然后就可以使用 RMAN 将备份存放到 FRA 上。我们将在第 11 章中讨论备份。

3.1.3　是否应该使用 FRA

我们认为使用在 FRA 的想法是不错的。我们也希望将备份复制到一些其他的介质中，例如磁带，从而可以异地使用它们以进行灾难恢复(当发生真正的灾难时，灾难恢复计划就会变得非常重要)。

我们希望将 FRA 用于归档的重做日志，但也希望将归档的重做日志复制到多个位置

(更为具体地说，是复制到多个磁盘)。需要牢记的是，归档的重做日志对于数据库恢复至关重要，如果丢失一个归档的重做日志，则在该重做日志之后的所有其他重做日志就没有任何价值。因此，我们倾向于使用 FRA 和至少一个位于不同磁盘位置上的其他的归档日志目标来配置数据库。这意味着使用 LOG_ARCHIVE_DEST_n 参数来配置数据库，而使用 FRA 和另一个单独的文件系统来存储归档重做日志。

促使我们倾向于使用 FRA 的另外一个重要的原因是其存储空间配额机制。当前相当多的数据库服务器上运行多个数据库。前面讨论过关于一个数据库将所有归档重做日志的物理磁盘空间消耗殆尽的情况。发生这种情况不仅对填满归档重做日志目标目录的数据库本身有影响，而且会影响系统上的其他所有数据库。采用配额系统，能够限制某数据库影响其他数据库。

有关我们希望涉及的内容有很多，例如备用数据库等方面，但这并不是本书所要介绍的内容。只要记住，底线是尽到保护数据的责任，因为没有什么比星期一早上开始工作但发现系统在周末时已经崩溃，而整个数据库以及所有的备份都已经丢失更糟的了。

3.1.4　ARCHIVELOG 模式之间的切换

一旦将数据库配置为在 ARCHIVELOG 模式下运行，则可以非常容易地实现数据库在 NOARCHIVELOG 模式和 ARCHIVELOG 模式之间的切换。为了将数据库置于 ARCHIVELOG 模式，必须首先使用如下的一种命令以一致的方式关闭数据库：shutdown 命令、shutdown immediate 命令或 shutdown transactional 命令。一旦正常地关闭了数据库，就可以执行 startup mount 命令来加载这个数据库。一旦加载了数据库，就可以执行命令 alter database archivelog 来将数据库置于 ARCHIVELOG 模式。然后可以使用 alter database open 命令打开该数据库。

如果希望数据库跳出 ARCHIVELOG 模式，则可以采用相反的过程。首先关闭数据库。一旦关闭了数据库，则通过执行 startup mount 命令加载数据库。一旦加载了数据库，则执行 alter database noarchivelog 命令将数据库置于 NOARCHIVELOG 模式。然后可以使用 alter database open 命令打开该数据库。

3.1.5　使用 Oracle 数据库配置助手创建数据库

如果使用 Oracle 数据库配置助手(Oracle Database Configuration Assistant，ODBCA)创建数据库，则很可能 Oracle 已经配置了大部分的 RMAN。ODBCA 将在 ARCHIVELOG 模式中配置数据库，配置 FRA，以及安排 RMAN 备份的调度。对于较小的安装软件，这可能就是需要完成的所有工作，并且不需要担心任何其他的基本 RMAN 配置问题。尽管如此，了解 RMAN 提供的所有选项仍然是必要的。例如，当使用 ODBCA 创建数据库时，不会启用备份的加密，并且您可能需要启用该特性。

RMAN 实习：将数据库置入 ARCHIVELOG 模式

实习注意事项

为了完成这个 RMAN 实习，您需要安装 Oracle 软件，并且在 NOARCHIVELOG 模

式下启动和运行一个数据库。在开始这个实习前，确定闪回恢复区的位置还要确定 Oracle 第二个归档日志目标目录的位置，该练习将归档到两个不同的位置。

步骤 1 为归档的重做日志配置 FRA 和单独的归档日志目标。首先，设置 FRA 参数 DB_RECOVERY_FILE_DEST_SIZE 和 DB_RECOVERY_FILE_DEST:

```
SQL> alter system set db_recovery_file_dest_size=2G;
System altered.
SQL> alter system set
db_recovery_file_dest='c:\oracle\product\10.2.0\flash_recovery_area';
System altered.
```

步骤 2 现在，定义两个归档日志目标目录，其中一个是 FRA。设置数据库参数文件并设置 LOG_ARCHIVE_DEST_1 参数，让它指向预先定义的文件系统，该文件系统将是第一个归档日志目录。由于要配置 LOG_ARCHIVE_DEST_1 参数目使用 1=RA，因此需要使用 USE_DB_RECOVERY_FILE_DEST 参数来设置 LOG_ARCHIVE_DEST_10 参数以指向 FRA。使用 show parameter 命令来验证该设置是否正确:

```
SQL> alter system set log_archive_dest_1='location=d:\archive\rob10R2';
System altered.
SQL> alter system set
log_archive_dest_10='LOCATION=USE_DB_RECOVERY_FILE_DEST';
SQL> show parameter log_archive_dest
NAME                          TYPE        VALUE
----------------------------- ----------- --------
log_archive_dest_1            string
location=d:\archive\rob10R2
log_archive_dest_10           string
LOCATION=USE_DB_RECOVERY_FILE_DEST
```

步骤 3 关闭数据库:

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

步骤 4 加载数据库:

```
SQL> startup mount
ORACLE instance started.
Total System Global Area    84700976 bytes
Fixed Size                    282416 bytes
Variable Size               71303168 bytes
Database Buffers            12582912 bytes
Redo Buffers                  532480 bytes
Database mounted.
```

步骤 5 将数据库置入 ARCHIVELOG 模式:

```
SQL> alter database archivelog ;
```

```
Database altered.
```

步骤 6　打开数据库：

```
SQL> alter database open;
Database altered.
```

虽然不是该实习的一部分，但这里给出的将这个数据库从 ARCHIVELOG 模式切换到 NOARCHIVELOG 模式的过程，即该实习中所描述过程的一个简单的逆过程：关闭数据库；执行 startup mount 命令来重新启动这个数据库实例；执行命令 alter database noarchivelog 将该数据库置入 NOARCHIVELOG 模式。需要注意，从 ARCHIVELOG 模式切换到 NOARCHIVELOG 模式时不要求以一致的方式关闭数据库。下面是一个将数据库切换回 NOARCHIVELOG 模式的示例：

```
SQL> shutdown
ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.
Total System Global Area    84700976 bytes
Fixed Size                    282416 bytes
Variable Size               71303168 bytes
Database Buffers            12582912 bytes
Redo Buffers                  532480 bytes
Database mounted.
SQL> alter database noarchivelog;
Database altered.
SQL> alter database open;
Database altered.
```

只要完成了上述的任一种切换任务，我们都应当立即备份这个数据库。

3.2　Oracle Database 11g 故障诊断基础架构

故障诊断基础架构(Fault Diagnosability Infrastructure)是 Oracle Database 11g 重要的新特性之一。贯穿本书各个章节，将涵盖与新的故障诊断有关的各种特性。该基础架构用于防止、检测、诊断、解决诸如类似数据库 bug 以及各种形式的崩溃等问题。新基础架构对以前的系统作了一些改变，例如建立警报日志的位置，并为 Oracle 数据库增加了许多新的功能。本书将详细讨论故障诊断基础架构，并讨论为数据库各类问题提供自动响应的支持工作台。第 13 章讨论数据库健康检查器(health checker)，是与 Oracle Automatic Diagnostic Repository(ADR)有关联的一个新组件。在第 12 章和第 13 章中，还将讨论数据库恢复顾问的使用问题，在数据库恢复期间，该问题与 ADR 密切相关。

出于为 Oracle 数据库设置故障诊断基础架构的目的，本章将关注新的参数 DIAGNOSTIC_DEST 的设置问题。新的参数 DIAGNOSTIC_DEST 定义了 ADR 根目录，并且不鼓励使用参数 USER_DUMP_DEST，参数 CORE_DUMP_DEST 和参数 BACKGROUND_ DUMP_

DEST 等参数。结果是，如果使用 DBCA 建立新的 Oracle Database 11*g* 数据库，将无法找到以前存在的警报日志和用户跟踪文件。

默认情况下，DIAGNOSTIC_DEST 参数被设置为$ORACLE_BASE。如果$ORACLE_BASE 未被设置，则设置为$ORACLE_HOME 的值。ADR 目录结构的根目录结构以名为 diag 的目录开始，其下是表示产品类型的子目录。例如，对于数据库，其产品类型称为 rdbms。在 rdbms 下是每个数据库的目录。数据库目录下是各个独立的实例。

例如，如果$ORACLE_BASE 的目录为/u01/oracle，数据库名为 mydb，数据库实例名为 mydb1，则该数据库的 ADR 目录结构为/u01/oracle/diag/rdbms/mydb/mydb1。该目录结构称为 ADR 主目录，不同的实例都有其自己的 ADR 主目录。若使用 RAC，则 ADR 可以使用共享存储方式，也可以使用各个节点独立存储的方式。如果处于 RAC 环境中，建议使用共享存储方式，采用该方式可以将各个节点的诊断数据聚集到共享存储中。一个共享的 ADR 能够为 Data Recovery Advisor 提供更加强大的恢复选项。

在该目录结构中包含另外一些目录，最常用的目录包括：

- **Alert 目录**　该目录为 XML 格式的警报日志位置。
- **cdump 目录**　该目录数据库核心转储位置。
- **Trace 目录**　该目录包括系统建立的跟踪文件，以及警报日志的文本副本。
- **Incident 目录**　该目录包含多个子目录，每个子目录包含一个事件。

图 3-1 所示为 ADR 目录的基本架构图。

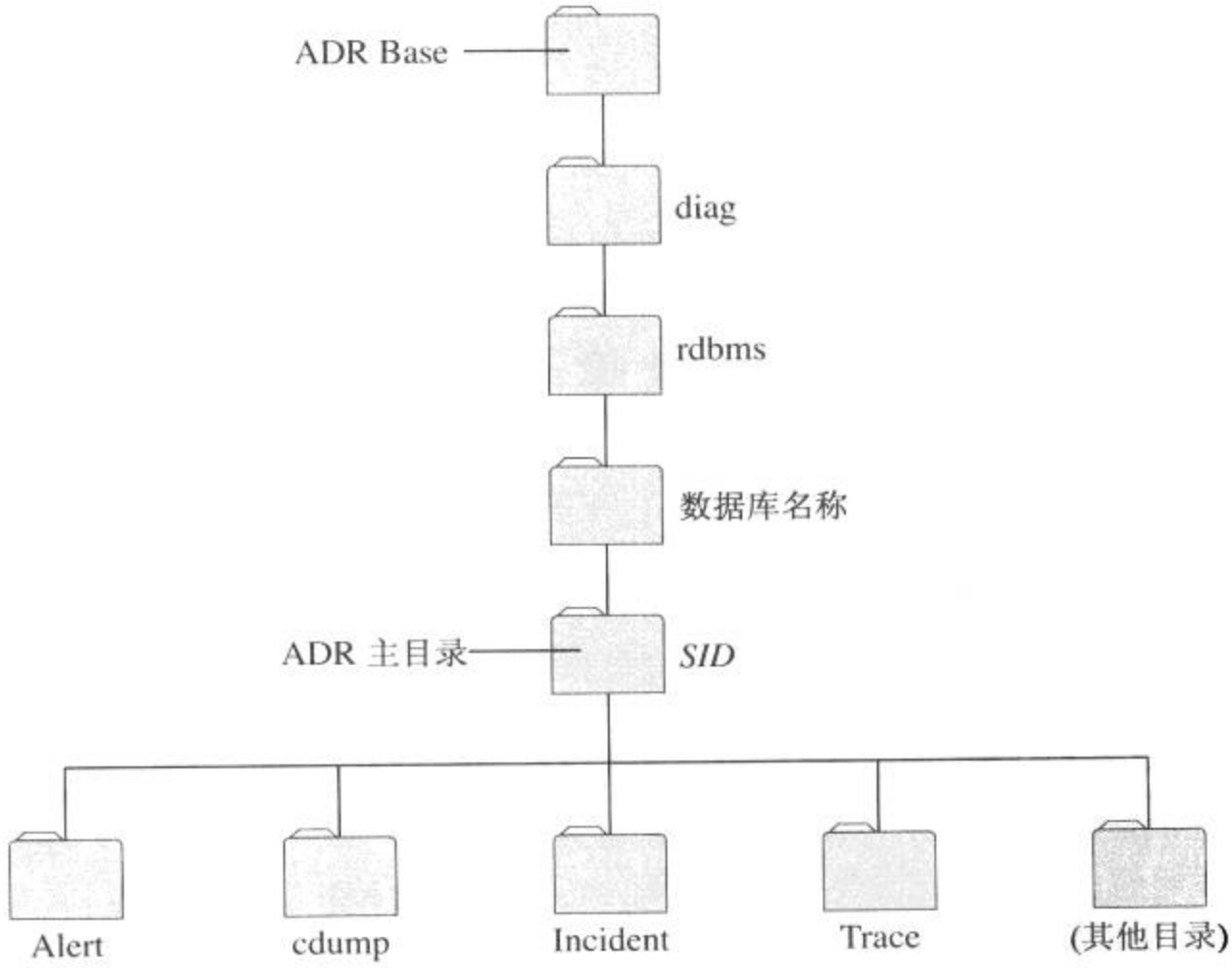


图 3-1　ADR 目录基本架构图

各种 ADR 位置的信息，以及 ADR 有关联的其他信息(如活动的事件)都可以通过新的视图 V$DIAG_INFO 获得。以下是通过 V$DIAG_INFO 视图查询的示例。

```
SQL> select * from v$diag_info;
   INST_ID NAME                 VALUE
---------- -------------------- --------------------------------
         1 Diag Enabled         TRUE
         1 ADR Base             C:\ORACLE\PRODUCT
         1 ADR Home             C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\
                                rob11gr4
         1 Diag Trace           C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\
                                rob11gr4\trace
         1 Diag Alert           C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\
                                rob11gr4\alert
         1 Diag Incident        C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\
                                rob11gr4\incident
         1 Diag Cdump           C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\
                                rob11gr4\cdump
         1 Health Monitor       C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\
                                rob11gr4\hm
         1 Default Trace File   C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\
                                rob11gr4\trace\rob11gr4_ora_7832.trc
         1 Active Problem Count  1
         1 Active Incident Count  1
11 rows selected.
```

3.3　RMAN 命令行

如果要执行联机备份操作，数据库应当处于 ARCHIVELOG 模式下，这时要准备配置 RMAN 和数据库，以便于备份操作。不过，最好先了解使用 RMAN 可执行文件的方法。下面将暂时放下有关设置的讨论，简单地介绍 RMAN 命令行接口(CLI)及其使用方法。

进入 RMAN 有两种不同的方法：一种是通过命令行，另一种是使用 OEM。第 11 章将详细介绍 OEM 接口的有关知识。本书中的大多数示例是使用命令行接口完成的，如果能够通过命令行进入 RMAN，其他方法都将是轻而易举的。在下一节中，将介绍如何使用 RMAN 命令行连接数据库以及如何使用 connect 命令。

3.3.1　通过 RMAN 命令行连接

在 OS 提示符下简单地输入 rman 命令就可以启动 RMAN。一旦启动了 RMAN 命令解释程序，就可以执行任何所需的操作。使用命令行参数完成一些预备工作通常会非常方便，因此可以在启动 RMAN 时传递一些命令行参数。可以使用这些命令行参数来使 RMAN 连接到需要备份的数据库(即目标数据库)、恢复目录或其他一些任务。表 3-3 提供了一个命令行参数、参数变量数据类型(如果存在)和参数用途的清单。

表 3-3 RMAN 命令行参数

RMAN 命令行参数	参数变量类型	用 途
target	字符串	定义要连接的目标数据库的用户名、口令和服务器名
catalog	字符串	定义恢复目录的用户名、口令和服务器名
nocatalog	不存在变量	表示会话不使用恢复目录，这是 *Oracle 8i* 和 *Oracle 9i* 中的默认参数
cmdfile	字符串	表示要执行的命令文件脚本名
log	字符串	表示 RMAN 会话应当记入日志，日志文件名为该参数的变量；还要删除显示屏上的所有 RMAN 消息(除了 RMAN 提示符之外)
trace	字符串	表示应当跟踪 RMAN 会话，跟踪文件名为该参数的变量
append	不存在变量	表示应当追加的日志文件(由日志参数定义)
debug	多种变量	表示应当以调试模式启动 RMAN
msgno	不存在变量	表示应当显示每种错误消息的 RMAN 前缀。如果没有选中这个选项，某些非错误消息就不会包含消息号
send	字符串	将字符串消息发送至介质管理层
pipe	字符串	调用 RMAN 管道接口
timeout	整数	表示等待管道输入的时间(以秒为单位)
Auxiliary	字符串	定义连接的辅助数据库的用户名、口令和服务器名
checksyntax	None	检查命令文件以列出语法错误
salxdebug	None	命令行及 RMAN 提示语法分析错误检查

下面是一些使用命令行参数启动 RMAN 的示例(后面还将介绍其他的示例)：

```
RMAN target=system/manager@robt nocatalog
RMAN target='sys/robert as sysdba@robt'nocatalog
RMAN target=system/manager@robt
catalog=system/manager@catalog log="RMAN.log"
RMAN target system/manager@robt nocatalog log "RMAN.log"
```

注意：
命令行参数和参数值之间的“=”号是可选的。另外，如果运行 Oracle Database 11g Real Application Clusters 软件，就只能连接群集中的一个实例。

注意，RMAN 总是以 SYSDBA 身份连接到目标数据库。了解这一点很重要，因为这暗示连接的账户必须具有 SYSDBA 权限。

如果忘记了与 RMAN 有关的命令行变量，还有一种方法可以显示有效的命令行参数：简单地使用一个无效的参数来启动 RMAN。在下面的方法示例中，我们可以看到

RMAN 会返回一个错误，不过同时也会提供一个有效命令行参数的列表(下面省略了列表末尾的一些错误)：

```
C:\Documents and Settings\Robert>rman help
    Argument      Value             Description
    ---------------------------------------------------------------------
    target        quoted-string     connect-string for target database
    catalog       quoted-string     connect-string for recovery catalog
    nocatalog     none              if specified, then no recovery catalog
    cmdfile       quoted-string     name of input command file
    log           quoted-string     name of output message log file
    trace         quoted-string     name of output debugging message log file
    append        none              if specified, log is opened in append mode
    debug         optional-args     activate debugging
    msgno         none              show RMAN-nnnn prefix for all messages
    send          quoted-string     send a command to the media manager
    pipe          string            building block for pipe names
    timeout       integer           number of seconds to wait for pipe input
    checksyntax   none              check the command file for syntax errors
    ---------------------------------------------------------------------
    Both single and double quotes (' or ") are accepted for a quoted-string.
    Quotes are not required unless the string contains embedded white-space.
```

RMAN 提供了 checksyntax 参数，它允许用户对有问题的 RMAN 命令进行检查。下面是使用 checksyntax 参数的示例：

```
C:\Documents and Settings\Robert>rman checksyntax
    Recovery Manager: Release 11.2.0.1.0 - Production on Thu Nov 5 04:03:03 2009
    Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
    RMAN> backup database pls archivelog;
    RMAN-00571: ===========================================================
    RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
    RMAN-00571: ===========================================================
    RMAN-00558: error encountered while parsing input commands
    RMAN-01009: syntax error: found "identifier": expecting one of: "archivelog,
    auxiliary, backupset, backup, channel, controlfilecopy, copy, current,
    database,
    datafilecopy, datafile, db_recovery_file_dest, delete, diskratio,
    filesperset,
    force, format, from, include, keep, maxsetsize, noexclude, nokeep, not, plus,
    pool, recovery, reuse, section, skip readonly, skip, spfile, tablespace,
    tag, to,
    (, ;"
    RMAN-01008: the bad identifier was: pls
    RMAN-01007: at line 1 column 17 file: standard input

    RMAN> backup database plus archivelog;
    The command has no syntax errors
```

注意，许多人都可能不仔细查看 RMAN 错误消息。在该消息中通常可以看出 RMAN 期望出现特定的关键字或短语。

3.3.2 RMAN 客户端兼容性

在使用 RMAN 客户端时，应该考虑该客户端与所连接的目标数据库的兼容性问题。还需要考虑恢复目录的兼容性(该问题将在第 9 章详细讨论)。一般来说，RMAN 客户端的版本应该与所连接的目标数据库版本相同或更高。表 3-4 所示的列表对目标数据库与辅助数据库及 RMAN 客户端之间的 RMAN 的兼容性问题提供参考和指导。

表 3-4 目标数据库、辅助数据库和 RMAN 客户端之间的 RMAN 兼容性列表

目标数据库/辅助数据库	客户端版本要求
8.0.6	只有 8.0.6 适用
8.1.7	只有 8.0.6.1 或 8.1.7 适用
8.1.7.4	只有 8.1.7.4 适用
9.0.1	只有 9.0.1 适用
9.2.0	≥9.0.1.3 和≤目标数据库可执行版本
10.1.0	≥9.0.1.3 和≤目标数据库可执行版本
10.2.0	≥9.0.1.3 和≤目标数据库可执行版本
11.1.0	≥9.0.1.3 和≤目标数据库可执行版本
11.2.0	≥9.0.1.3 和≤目标数据库可执行版本

RMAN 存储在$ORACLE_HOME/bin 目录中。该目录应该在 ORACLE 运行的操作系统的路径上。若存在多个 ORACLE_HOME 目录就应该引起注意。在启动 RMAN 前应确保操作系统路径指向正确的 ORACLE_HOME 目录。如果路径设置不正确，则会造成使用错误的 ORACLE_HOME 目录。另外还需要注意在 RMAN 执行前没有其他的 rman 命令正在运行。例如，某种已存在路径中的 Unix 版本的 rman 命令正在运行而不是 RMAN 正在运行，因为它在路径的首位。若存在这样的情况，则需要对路径进行调整，或者将其改变为$ORACLE_HOME/bin 目录，直接从该位置运行 rman 命令。

通常来说，如果使用不兼容的客户端版本，则 RMAN 会产生一个错误。可以使用操作系统级实用程序(如 oraenv)检查使用的目录是否是正确的 ORACLE_HOME 目录，或者检查 RMAN 客户端的标志。

3.3.3 使用 RMAN 的 connect 命令

如果启动了 RMAN，发现没有连接到正确的数据库，或者是要连接一个不同的数据库(目标数据库、目录数据库或辅助数据库)，可以使用 connect 命令来更改 RMAN 要连接的数据库。使用 connect target 命令可以连接到一个不同的目标数据库；使用 connect catalog 命令可以连接到一个不同的恢复目录；使用 connect auxiliary 命令可以连接到一个

不同的辅助数据库。下面是一些使用 connect 命令的示例：

```
connect target sys/password@testdb;
connect catalog rcat_user/password@robdb;
```

3.3.4　退出 RMAN 客户端

当使用完 RMAN 时，有必要退出 RMAN 客户端。RMAN 提供了两个命令：quit 和 exit。使用这些命令可返回到 OS 提示符。RMAN 也允许使用 host 命令返回到 OS。下面是一些示例。

```
C:\>rman target=/
Recovery Manager: Release 11.2.0.1.0-Production on Wed Oct 4 22:49:14 2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.
connected to target database: ORCL2 (DBID=582838926)
RMAN> host;
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\>exit
host command complete
RMAN> exit
Recovery Manager complete.
```

3.4　为 RMAN 操作配置数据库

现在我们已经了解了启动 RMAN 的方法，接下来就需要处理某些配置项。虽然只是启动 RMAN 也可以执行备份操作，但在这样做之前，最好能够先解决一些配置问题。首先，需要设置要使用 RMAN 的数据库用户；随后还要配置 RMAN 使用的一些默认设置，所以还将介绍这些默认设置。

3.4.1　设置数据库用户

在默认情况下，可以通过 SYS 账户(即 sysdba)来使用 RMAN，该账户不需要任何配置。当然，在执行产品备份操作时，sysdba 并不是最佳的账户。建议在使用 RMAN 执行备份操作之前创建一个用于 RMAN 备份的单独账户设置。操作如下所示。

RMAN 实习：创建目标数据库 RMAN 备份账户

实习注意事项

为了完成这个 RMAN 实习，需要安装 Oracle 软件，并且启动和运行一个数据库。您需要拥有这个数据库的管理权限。

步骤 1　确定要使用的用户账户名，使用数据库 create user 命令创建这个账户：

```
CREATE USER backup_admin IDENTIFIED BY backupuserpassword
default TABLESPACE users;
```

步骤2 授予BACKUP_ADMIN用户sysdba权限，这是因为RMAN会经常使用sysdba登录来连接数据库。下面是一个授予BACKUP_ADMIN账户sysdba权限的示例：

```
GRANT sysdba TO backup_admin;
```

使用Oracle数据库配置助手创建的数据库

如果使用dbca创建数据库，则有一个选项可用来设置自动执行的每日备份。如果选择这个选项，Oracle将执行一些初始的RMAN备份(例如，配置FRA)。虽然这种RMAN配置对于不重要的数据库已经足够，但如果正在管理的是至关重要的数据库，则应该遵循本章中前面介绍的步骤，并确保正确配置了数据库。

如果试图将RMAN与一个没有正确创建的账户连接，就会产生下面的错误：

```
D:\oracle\oradata\robt>RMAN target=backup/backup@robt
Recovery Manager: Release 11.2.0.1.0
Production on Tue Aug 22 21:40:51 2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-00554: initialization of internal recovery manager package failed
RMAN-04005: error from target database:
ORA-01031: insufficient privileges
```

现在已经创建了账户并授予其所需的权限，然而要使用RMAN，还必须配置一些RMAN的默认设置。下面将介绍这些默认设置。

3.4.2 设置数据库安全性

下面简要讨论在本地服务器上连接RMAN与通过Oracle Net连接RMAN两者之间的差异。启动RMAN时，可能登录到数据库所在的服务器。在这种情况下，如果使用特许的OS用户账户登录，只需要执行前面的RMAN实习中的步骤1和步骤2即可。根据使用的操作系统，可以确定我们的用户账户是否是特许的。如果使用UNIX系统，通常会存在一个名为dba(名字有可能不同)的UNIX组，它是在创建Oracle特有的账户(通常名为Oracle)时创建的。如果在这个UNIX组中指定我们的UNIX用户账户，就可以不再执行其他操作而连接到一个目标数据库。如果使用Windows平台，特许用户会指定到一个通常名为ORA_DBA的NT组。

如果没有使用特许的账户登录本地服务器，或者从客户端工作站使用Oracle Net来连接目标数据库(例如使用system/manager@testdb来连接)，就需要配置数据库使用口令文件。配置口令文件时，首先要创建口令文件，然后要配置数据库，使数据库明确使用这个口令文件。下面将详细介绍这两个步骤。

1. 创建口令文件

使用Oracle实用程序orapwd可以创建数据库口令文件，该命令具有3个参数。

- file 参数：口令文件名。
- password 参数：sys 用户的口令。
- entries 参数：为其他特许的 Oracle 用户账户保留的条目数。

在默认情况下，NT 上的 Oracle 数据库对口令文件采用这样的命名标准：PWD*sid*.ora，其中的 sid 就是数据库名。下面是创建口令文件的示例：

```
orapwd file=PWDrobt.ora password=robert entries=20
```

创建了口令文件后，接下来需要配置数据库，以让它使用口令文件，以便可以通过 Oracle Net 来执行远程备份操作。

2. 配置数据库使用口令文件

在默认情况下，Oracle 数据库不会被配置为使用口令文件(除非使用 ODBCA 创建数据库)。要配置数据库，可以在常用的编辑器中编辑参数文件(init.ora)。我们要编辑的参数为 REMOTE_LOGIN_PASSWORDFILE，在 Oracle Database 11g 中该参数可以设置为下面的 3 个参数中的一个：

- none 参数：默认值。在这种情况下，Oracle 会忽略口令文件，并且 sysdba 访问只给予本地特许的登录。
- shared 参数：该参数说明多个数据库可以使用相同的口令文件。这种模式只能保存 SYS 用户账户的口令。
- exclusive 参数：该参数说明只有一个数据库可以使用这个口令文件。在这种模式中，口令文件可以包含用于多个特许的 Oracle 账户的口令。这是推荐的操作模式，特别是在运行 RMAN 时。如果希望将 RMAN 与来自于远程客户端的数据库连接，则必须使用该参数设置。

如果使用的是 Oracle Database 的 SPFILE 而不是基于文本的参数文件，则使用 alter system 命令可以更改这个参数的设置，如下所示：

```
alter system set REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE scope=spfile;
```

最后需要说明的是，REMOTE_LOGIN_PASSWORDFILE 参数不是可动态改变的，所以不能在数据库启动和运行时更改这个参数。相反，需要更改 SPFILE(使用 alter system 命令的 scope=spfile 参数更改)，然后关闭并重启数据库。

3.4.3　设置 CONTROL_FILE_RECORD_KEEP_TIME 参数

为 RMAN 配置数据库时，应当考虑在控制文件中存储备份记录的时间。备份记录包括完全的数据库备份记录，以及指定的数据文件、控制文件、参数文件和归档目录的备份记录。数据库参数 CONTROL_FILE_RECORD_KEEP_TIME 以天为单位(默认值为 7)，因此在默认情况下，Oracle 会将 RMAN 备份和恢复记录保留 7 天。可以将该参数设置为 0 到 365 之间的任意值。

CONTROL_FILE_RECORD_KEEP_TIME 参数会影响一系列数据库操作。首先，产

生 RMAN 备份时，由于与这些备份相关的记录存储在控制文件中，所以该参数直接影响数据库控制文件的大小。备份记录将不断地保存在控制文件中，控制文件将耗尽空间。这时，Oracle 会扩展控制文件来调整备份记录所需的存储空间。此外，该参数设为 0 时将禁止扩展控制文件，并会使得 RMAN 备份的保存周期不确定。

我们建议将 CONTROL_FILE_RECORD_KEEP_TIME 参数值设置为不小于选中数据库的备份保存周期的值，否则有可能在备份介质上有数据库备份，但在控制文件中不存在与备份相关的备份记录。如果由于某些原因需要恢复这些较早的记录，就会导致严重的问题。

注意：
如果在某些位置上错误设置了文件保存策略，则会导致备份保存策略失败。这包括错误设置了 CONTROL_FILE_RECORD_KEEP_TEME 参数，RMAN 保存策略，及磁带供应商产品提供的保存策略。确保正确设置了保存策略，以保证能正常地还原备份。

3.5　配置 RMAN 默认设置

RMAN 允许执行自动的数据库备份和恢复，后面将详细讨论这个问题。为了支持这个功能，RMAN 允许为一些设置定义默认值(如信道配置)。本节将介绍 RMAN 默认设置的配置。如果能够配置这些设置，就意味着可以更改这个配置，有必要的话甚至可以完全删除这个配置。所以这一节也会介绍更改和删除配置的内容。所有这些配置工作都是为了方便执行备份操作的进程。首先，我们会大致介绍 RMAN 中的 configure 命令和该命令提供的所有功能，随后还将介绍可能需要使用 configure 命令配置的一些默认值。

我们将在本节中使用一些您可能还不熟悉的术语，后面的章节将讲述这些术语。虽然您对有些术语的概念比较模糊，但是许多术语已经在第 2 章中介绍过。对于 RMAN 来说，没有任何默认配置选项是必需的。我们建议您先粗略阅读一下本节，对能够设置的多种默认值有一个基本的认识，然后在学习和应用后面的章节后再重新阅读这一节。届时，您就能很容易地判断要采用 Oracle 数据库的哪些默认设置。

3.5.1　configure 命令介绍

RMAN 提供了 configure 命令，该命令允许用户为备份和恢复操作定义默认值。RMAN 还允许用户使用 configure 命令来更改多种参数的默认值，这些参数值会持续到再次被清除或被更改。自定义默认配置选项的能力允许我们执行自动的 RMAN 操作。下面是可以配置的几个不同设置：

- 用于 RMAN 操作的默认设备类型，如磁盘或 SBT(系统备份磁带设备)。
- 执行自动的备份和还原操作时自动分配的信道数。
- 配置将指定表空间排除在完全数据库备份操作之外的表空间排斥策略。
- 自动备份时任何给定备份片的最大空间和任何备份集的大小。

- 备份优化的默认值为 ON 或 OFF。备份优化会删除同一数据文件(例如与只读表空间关联的数据文件)和归档的重做日志的复制备份。
- 快照控制文件的默认文件名(第 2 章介绍过快照控制文件)。
- 控制文件自动备份默认为 ON 或 OFF，控制文件备份的输出文件的默认格式，以及创建这些备份的默认设备也如此。
- 辅助数据库文件的默认文件名。
- 默认的保留策略，该策略用于确定哪些备份和副本符合删除条件。
- 默认的加密值和关联的加密算法。
- 当需要压缩时，使用的默认压缩算法。
- 归档重做日志的删除策略。

上面的每一种可配置设置都具有指定的默认值，这些默认值(与任何配置值一样)存储在数据库控制文件中。即使是连接到恢复目录，情况仍然如此。使用 show 命令可以查看各种 RMAN 参数的当前配置值。V$RMAN_CONFIGURATION 数据库视图还会列出非默认的 RMAN 配置设置。下面是一些使用 show 命令的示例：

```
show default device type;
show maxsetsize;
show retention policy;
show all;
```

3.5.2　配置多种 RMAN 默认设置

本节将介绍 RMAN 默认值的设置。首先，将介绍信道默认设置的配置。可以采用不同的方法来配置信道：使用 configure channel device type 命令可以配置用于所有信道的默认值；使用 configure channel *n* device type 命令可以配置用于指定默认信道的默认值。

使用 configure channel device type clear 命令可以清除用于所有信道的默认值，使用 configure channel *n* device type clear 命令可以清除用于指定默认信道的默认值。

使用 allocate channel 命令分配一个信道时，可以为分配的这个信道指定名称。例如，allocate channel d1 device type disk 命令会创建一个名为 d1 的信道。自动分配信道时，Oracle 会为这些信道指定名称，它们的默认名称取决于其默认设备类型，如表 3-5 所示：

表 3-5　信道的默认名称及示例

设 备 类 型	默认名格式	示　例
磁盘	ORA_DISK_*n*	ORA_DISK_1，ORA_DISK_2
磁带	ORA_SBT_TAPE_*n*	ORA_SBT_TAPE_1，ORA_SBT_TAPE_2

自动分配的信道数取决于所定义的默认并行级别(本章后面部分将做介绍)。

执行 configure 命令时，Oracle 显示先前的配置设置，然后显示新的配置设置。下面介绍使用 configure 命令来自动执行 RMAN 备份和还原进程的一些方法。

3.5.3　configure 命令的使用示例

本节介绍一些使用 configure 命令定义默认值的示例。在本节中将介绍一些有关 config 命令的主题，包括：

- 配置信道默认设置
- 使用格式串
- 配置默认的控制文件自动备份和 SPFILE 自动备份
- 配置默认的保留策略
- 配置默认的加密级别
- 默认归档日志删除策略

1. 配置信道默认设置

首先要介绍的是将默认的备份/还原设备配置为磁带或磁盘的示例。在该示例中，为备份操作指定的所有信道都分配到磁盘上：

```
CONFIGURE DEFAULT DEVICE TYPE TO SBT;
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
```

当配置默认的设备类型时，信道除非使用 backup device type 参数重写默认值，Oracle 将使用默认的信道。也将自动分配用于 delete 命令的维护信道和用于复制操作的辅助信道。

一旦配置了默认的设备类型，就可以配置特定备份类型的默认值，当使用设备时应该进行这种备份。例如，当备份到磁盘时，可以让 Oracle 默认地使用标准的 Oracle 备份集方法备份数据库，或者可以让其默认为使用副本(只可以备份到磁盘)。也可以默认为压缩备份集，并表明并行程度(这表示将分配给备份的信道数)。下面是这些不同选项的配置示例：

```
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO BACKUPSET;
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COMPRESSED BACKUPSET;
CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY;
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
```

压缩是 Oracle Database 10g 中 RMAN 提供的新特性。压缩提供了 Oracle 备份集的真正压缩，而不同于 ZIP 压缩。它可以有效地减小备份集的大小。当然，压缩将占用资源，并且需要更长的时间来完成备份和还原。

现在再来看一个在自动备份或恢复操作期间配置要分配的信道数的示例。同样，在这个示例中，我们设置磁盘操作的默认并行级别为 2，这样，如果启动自动的备份操作，就会分配两个信道来并行执行备份操作，如下所示。

```
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT 'd:\backup\robt\backup_%U';
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT 'e:\backup\robt\backup_%U';
```

注意：
通常在设置默认的并行级别时，应当设置其为连接备份数据库的磁盘数或磁带驱动器数。在使用磁盘时，需要通过反复试验确定。因为磁盘有多个磁头，且可能会卸下，由此使用多信道可能会获得更好的吞吐率。由此需要对实际使用磁盘并行性能进行测试，并根据测试结果设置。

在配置信道时有一些可用的选项。使用 maxpiecesize 参数，可以控制备份集片的大小。使用 maxopenfiles 参数，可以控制 RMAN 一次最多可以打开的文件数。rate 参数用于限制 RMAN 并控制备份时读取字节的比率为 B/S、KB/S、MB/S 或 GB/S。

下面的示例综合使用所有这些选项。我们限定信道 1 创建的每个备份片最大为 100MB，并且限定 RMAN 在该信道上最多打开 8 个文件。最后，信道被限制为只允许拥有 100MB 以内的吞吐量。

```
CONFIGURE CHANNEL 1 DEVICE TYPE DISK MAXPIECESIZE 100m maxopenfiles 8
    rate 100MB;
```

注意：
不要混淆 maxpiecesize 参数和 maxsetsize 参数之间的区别：maxpiecesize 限定单个备份集片的大小，该参数对备份的整体大小没有影响。另一方面，maxsetsize 参数可以限定备份的整体大小，因此需要谨慎使用。

如果要限定所有信道，我们执行的命令会稍有不同，该命令如下所示：

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 100m;
```

为什么需要更改给定备份集片的最大尺寸呢？首先，某些指定文件的大小存在限制。例如磁带只能处理一定量的数据，而某些磁盘文件系统对给定的数据文件的大小有一些限制。

此外，还可能要将一个磁带设备设置为所有信道的默认设备，并且指定一些参数的设置。在这种情况下，configure 命令可能会如下所示：

```
-- Note that we could have used the = sign after the PARMS clause if
-- we preferred like this:
-- PARMS='ENV=(NB_ORA_CLASS=RMAN_rs100_tape).
-- This is true with many parameters.
CONFIGURE CHANNEL DEVICE TYPE sbt MAXPIECESIZE 100m
PARMS 'ENV=(NB_ORA_CLASS=RMAN_rs100_tape)';
```

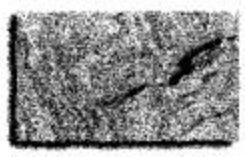
使用 configure 命令时，可能会发现需要清除给定的配置，这样才能使用默认值。使用 configure 命令的 clear 选项可以完成这一操作。在下面的示例中，将清除为默认信道 1 设置的默认选项。

```
CONFIGURE CHANNEL 1 DEVICE TYPE DISK CLEAR;
```

2. 配置与备份集有关的设置

可能还要配置整个备份集的默认最大尺寸，这时只需对语法稍作修改(下面的示例把最大尺寸设置回默认值，即没有限制)：

```
CONFIGURE MAXSETSIZE TO 7500K;
CONFIGURE MAXSETSIZE CLEAR;
```



警告：
使用 maxsetsize 来限定创建的整个备份的大小时需要特别小心。这是因为数据库在初始阶段可能小于定义的 maxsetsize 值，不过数据库很快就会超过 maxsetsize 值，并会导致数据库备份操作失败。

在后面的章节中，可以将备份进程配置为创建双路备份(duplexed backup)，也就是说在不同的位置创建同一个备份的多个副本。还可以使用 configure 命令来配置数据库的默认设置，以使用 configure 命令执行双路的自动备份操作。

下面的示例定义了任何备份到磁盘上的备份都将被复制，拥有两个副本：

```
configure datafile backup copies for device type disk to 2;
```

在自动备份期间，可能要将指定的表空间排除在备份之外，Oracle 允许使用 configure 命令来实现这一功能。下面的示例配置默认为排除一个表空间：

```
configure exclude for tablespace old_data;
```

configure 命令允许我们启用或禁止备份优化。启用备份优化时，它会使 Oracle 跳过在备份设备上已有相同备份的文件的备份。下面是一个配置备份优化的示例：

```
configure backup optimization on;
```

需要注意的是，要执行优化操作，就必须启用备份优化。此外，我们也必须执行 backup database 命令(或 backup archivelog 命令)与 like 选项(或 all 选项)，或者是使用 backup backupset all 命令(后面章节将详细介绍这些备份类型)。最后，还可以使用 backup 命令的 force 参数来禁止设置备份优化。

3. 配置快照控制文件设置

第 2 章介绍过快照控制文件，该文件是 RMAN 备份操作期间数据库控制文件的时间点副本，这样可以确保备份与给定时间点一致。因此，如果启动备份操作(假定为一个联机备份操作)后在数据库中添加一个表空间或数据文件，这个表空间或数据文件就不会包含在备份中。有时候我们希望 RMAN 在默认位置以外的位置创建备份控制文件，这时可以使用下面的 configure 命令为快照控制文件定义一个新的默认位置。

```
configure snapshot controlfile name to
'd:\oracle\backup\scontrolf_mydb';
```

注意，即使配置了 FRA，Oracle 也不会在 FRA 中创建快照控制文件。注意上面的示例，它包含数据库名(如果运行在 RAC 环境可能是数据库实例)，以此方法确保快照控制文件名是唯一的。

4. 使用格式串

注意在前面的示例中，在一些位置中定义了一个或多个磁盘位置和文件名格式。这称为格式串规范。本书中将大量地使用格式串规范，而在使用 RMAN 时也会经常用到格式串，除非使用 FRA。FRA 使用 Oracle 本身的文件名命名规则，当备份到 FRA 中时，建议不要使用格式串(可能为文件维护带来问题)。由于 FRA 是备份的默认位置，由此不需要将备份设备指向 FRA。可能或有其他原因导致您需要配置信道，不要将它们配置为指向 FRA 的格式串。

格式串与平台无关(尽管目录结构与平台有关)。Windows 上的格式串与 Unix 或其他平台均基本相同。例如，如果使用 Unix 系统，格式串类似于如下所示：

```
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT
'/u01/opt/oracle/backup/robt/backup_%U';
CONFIGURE CHANNEL 2 DEVICE TYPE DISK FORMAT
'/u01/opt/oracle/backup/robt/backup_%U';
```


注意：

如果使用 format 参数，Oracle 将不会管理备份文件，即使是备份到 FRA，这是因为 Oracle 不会管理备份。如果使用 format 参数，则保留策略就必须删除格式化的备份。如果没有使用 format 参数，则采用 OMF 名称并在 FRA 中创建文件。当备份到 FRA 时，不要使用 format 选项。

在 configure 命令中会经常使用格式串。在 backup、restore 和 allocate channel 等其他 RMAN 命令中也会经常看到格式串。RMAN 提供了与格式串规范关联的一些语法元素。这些元素是占位符，RMAN 将使用相应的定义值替换它们。例如，前面示例中的%U 语法元素告诉 RMAN 要使用系统生成的唯一标识符替换文件名。%U 用于确保每个备份文件名的唯一。表 3-6 列出了有效的语法元素并提供了这些元素用法的简单说明。

表 3-6　格式串规范说明

元　素	说　明
%a	表示应该替换数据库的激活 ID
%b	定义没有任何目录路径的文件名。该定义只能被 set newname 命令使用，或者用于使用图像副本建立备份
%c	指定应该替换一组双路备份片中的备份片副本数量(最大值为 256)。数字为 1 表示非双路备份集，0 表示代理副本
%d	表示应该替换数据库的名称
%D	表示应该替换某个月的当前日期，格式为 DD

(续表)

元 素	说 明
%e	表示应该替换归档的日志的序列号
%f	表示应该替换绝对的文件数
%F	提供唯一的和可重复的名称，该名称结合了数据库 ID(DBID)、日期、月份、年份和顺序
%h	表示应该替换归档的重做日志线程数
%I	表示应该替换 DBID
%M	表示应该替换日历中的某个月，格式为 MM
%N	表示应该替换表空间名
%n	表示应该替换数据库名称，右边用 x 字符填充直到总长为 8 个字符。 例如，如果 ROBDB 是数据库名，则填充后的名称是 ROBDBxxx
%p	表示应该替换备份集中的片数量。对于每个备份集，该值最初为 1，在创建每个备份片时，该值增加 1
%s	表示应该替换备份集数量。该数量是控制文件中的计数器，根据每个备份集增加计数器的值。计数器值初始为 1。该数量将在控制文件的生命周期中唯一(因此，在执行 RESETLOGS 时复位该数量，或者在还原或重新创建控制文件时复位该数量)
%t	表示应该替换备份集时间标记，这是根据从固定参考时间以来已经过去的秒数得出的 4 位字节值。可以结合使用%s 和%t 来形成备份集的唯一名称
%T	表示应该替换格式为 YYYYMMDD 的年、月、日
%u	表示应该替换 8 位字符的名称，该名称由备份集或图像副本数量以及备份集或镜像副本的创建时间的压缩表示组成
%U	这是默认的文件命名模式，它为 RMAN 相关的文件提供了系统生成的唯一文件名。 当处理镜像副本或备份片时，这种替换为串的意义会有所不同。 当使用备份集片时，%U 代表%u_%p_%c 的缩写，%u_%p_%c 保证生成的备份文件名的唯一性。 当使用镜像副本时，意义会有所不同，并且会因镜像副本类型的不同也会有不同的意义。 用于数据文件的镜像副本时，意义如下： data-D-%d_id-%I_TS-%N_FNO-%f_%u 用于归档的重做日志的镜像副本时，意义如下： arch-D_%d_id-%I_S-%e_T-%h_A-%a_%u 用于控制文件的镜像副本时，意义如下： cf-D_%d-id-%I-%u
%Y	表示应该替换格式为 YYYY 的年份
%%	表示希望使用%字符；例如%%Y

5. 配置控制文件和数据库参数文件的自动备份

Oracle Database 10g 以上版本的 RMAN 提供备份控制文件和数据库参数文件的功能，可以配置默认执行这些备份操作。可以再次使用 configure 命令进行配置，使得在备份操作期间自动执行这个自动备份进程。下面的示例配置自动备份这些重要的数据库文件并关闭默认配置：

```
configure controlfile autobackup on;
configure controlfile autobackup off;
```

配置控制文件和参数文件的自动备份时，可以应用下面的规则：

- 在 run 程序块之外执行每个 RMAN 的 backup 或 copy 命令时，都会自动备份控制文件和服务器参数文件。
- 如果使用了 run 程序块而且该程序块的最后一条命令不是 backup 或 copy 命令，则在 run 程序块结尾处会备份控制文件和参数文件。

注意：

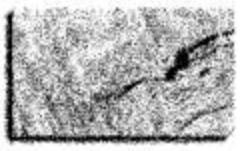
如果不打算使用恢复目录，则应该满足下列条件：

- 在控制文件自动备份后，希望能恢复控制文件
- 没有使用 FRA

除了上面这两种自动的控制文件备份类型，我们还可以配置另一种特殊的控制文件备份类型，该类型发生在数据库变化时，这些变化包括添加新的表空间、添加数据文件、添加联机重做日志等。这种自动备份类型只在磁盘上发生，使用 configure controlfile autobackup 命令的一种特殊选项可以简化这种备份。下面是一个示例：

```
RMAN> configure controlfile autobackup format for device type
disk to 'd:\backup\contf\robt_%F'
```

使用该选项时，Oracle RDBMS 会在数据库结构的变化对控制文件产生影响时自动备份控制文件。这些变化可能包括添加新的表空间，改变表空间或数据文件的状态(如使其联机)，添加新的联机重做日志，重命名文件，以及添加新的重做线程等。需要注意的是，这种自动备份只在磁盘上发生，它并不支持磁带操作。这些备份可能较大(这是因为控制文件包含许多老备份的历史记录)，因此必须确保为备份目录分配了足够大的磁盘空间。若不考虑所需的额外空间，这些备份对于恢复操作来说极为方便。最后注意，即使由于某种原因备份操作失败，数据库操作本身也不会失败。



注意：
如果不使用恢复目录而又希望在自动备份控制文件后能够恢复控制文件，就必须知道数据库的数据库 ID(DBID)。作为 RMAN 初始化设置和配置的一部分，我们应当记住要备份数据库的 DBID，并在安全的地方保存这个 DBID。可以从 V$DATABASE 视图的 DBID 列中获得数据库的 DBID。当启动 RMAN 并连接到目标数据库时，也会显示数据库的 DBID，如同下面的示例所示：

```
[oracle@robertgfreeman ~ ] $ rman target=/
Recovery Manager: Release 11.2.0.1.0 - Production on Thu Nov
5 04:040:6 2009
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.
connected to target database: ROB1 (DBID=1854903786)
```

6. 配置默认的保留策略

那么，数据库备份需要保留多长时间呢？RMAN 允许用户使用 configure retention policy 命令配置备份保留策略。如果在使用 FRA 的情况下配置了保留策略，则当备份过时后，ORACLE RMAN 将会自动删除这些过时的备份。若未使用 FRA, 配置的保留策略不会自动删除备份。在执行 report obsolete 命令时，将会列出到期的备份集。(第 15 章会详细介绍 RMAN 的 report obsolete 命令)。

Oracle 包含三类保留策略：基于窗口的恢复，基于冗余的恢复，无保留策略的恢复。下面将详细分析每一种恢复。

基于窗口保留策略的恢复 基于窗口保留策略的恢复可确保数据库能够恢复到特定的时间点。例如，如果希望将数据库恢复到三天前某个确定的时间点(当然，该数据库运行在 ARCHIVELOG 模式下)，就应该将基于窗口的保留策略恢复设置为 3 天。配置该保留策略的命令如下：

```
RMAN> configure retention policy to recovery window of 3 days;
    Old RMAN configuration parameters:
    CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
    new RMAN configuration parameters:
    CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 3 DAYS;
    new RMAN configuration parameters are successfully stored
```

需要注意的是即使采用基于窗口的保留策略，实际的备份存在的时间将比定义的保存时间长。例如，假定恢复窗口是 3 天，系统中最近的完整备份是 5 天前，该备份将会始终有效，直到不再需要该备份还原数据库为止。即使该备份是当天的备份，5 天后，这个 5 天前的备份仍然有效，因为该备份是将数据库系统恢复到 3 天前的状态的唯一可靠的备份。图 3-2 给出了图示化的说明。

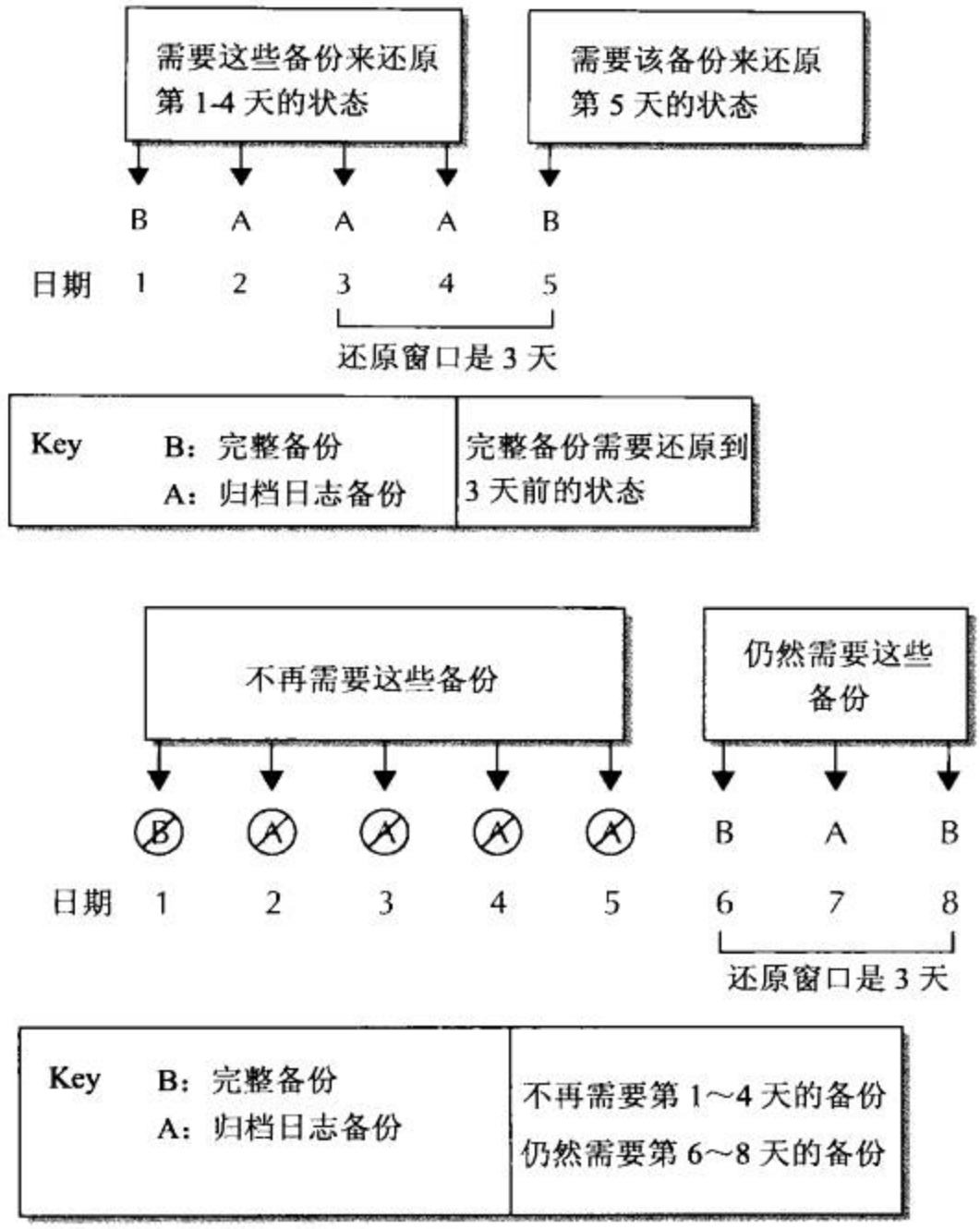


图 3-2　恢复窗口保持的旧的备份

配置完保留策略后，我们可以查看已生成的已作废的备份：

```
RMAN> report obsolete;
RMAN retention policy will be applied to the command
RMAN retention policy is set to recovery window of 3 days
Report of obsolete backups and copies
Type                 Key    Completion Time    Filename/Handle
-------------------- ------ ------------------ --------------------
Archive Log           12       08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08/
O1_mf_1_9_5bd8qv45_.arc
Backup Set            24       08-SEP-09
  Backup Piece        34       08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
O1_mf_annn_TAG20090908T202600_5bg4kr90_.bkp
Backup Set            25       08-SEP-09
  Backup Piece        35       08-SEP-09
```

```
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
o1_mf_nnnd0_TAG20090908T202601_5bg4ktk1_.bkp
```

该示例显示基于保留策略，有两个备份集和两个相关的备份片过期。另外，还有一个归档重做日志也需要删除。如果这些备份在定义的 FRA 中，Oracle 将按照需要删除它们。如果没有使用 FRA,或者是这些备份在尚未转换到使用 FRA 时已经建立，需要使用 delete obsolete 命令删除它们。在第 15 章中有更多关于 delete obsolete 命令的详细介绍。下面提供一个示例：

```
RMAN> delete obsolete;
RMAN retention policy will be applied to the command
RMAN retention policy is set to recovery window of 3 days
using channel ORA_DISK_1
using channel ORA_DISK_2
Deleting the following obsolete backups and copies:
Type                 Key    Completion Time Filename/Handle
-------------------- ------ ------------------ --------------------
Archive Log          12     08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08/
o1_mf_1_9_5bd8qv45_.arc
Backup Set           24     08-SEP-09
  Backup Piece       34     08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
o1_mf_annnn_TAG20090908T202600_5bg4kr90_.bkp
Backup Set           25     08-SEP-09
  Backup Piece       35     08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
o1_mf_nnnd0_TAG20090908T202601_5bg4ktk1_.bkp
Do you really want to delete the above objects (enter YES or NO)? yes
```

需要注意的是，在前面的示例中，系统会请求您确认是否删除这些对象。如果列出的某个对象不能被删除，就需要运行 crosscheck 命令(第 14 章将讨论该命令)。否则，Oracle 会删除 delete obsolete 命令输出中列出的所有项。

基于冗余的保留策略 如果很少备份数据库，就可能需要使用冗余策略。冗余策略的依据是备份数，而不是备份生成的天数。该策略是默认的保留策略，在系统中其值为 1.若将该值设置为 3，Oracle 将会保存最近三个备份，其他备份将被删除。示例如下：

```
RMAN> configure retention policy to redundancy 3;
old RMAN configuration parameters:
CONFIGURE RETENTION POLICY TO REDUNDANCY 3;
new RMAN configuration parameters:
CONFIGURE RETENTION POLICY TO REDUNDANCY 3;
new RMAN configuration parameters are successfully stored
```

注意该配置命令执行将显示过去的和新的设置结果。

无保留策略的恢复　最后，如果要禁止保留策略，可以使用 configure retention policy to none 命令，这样将不应用任何保留策略。使用 configure retention policy clear 命令可以将保留策略重置为默认值(冗余为 1)。

注意：
如果使用磁带管理系统，该系统会具有自己的保留策略。如果磁带管理系统的保留策略与 RMAN 中定义的备份保留策略冲突，则磁带管理系统的保留策略优先，同时恢复备份的能力会大受影响。

7. 配置默认的加密级别

在 Oracle Database 10*g* Release 2 和后面的版本中，RMAN 可以创建加密备份。在备份期间，Oracle 在创建备份集时对其进行加密。当还原备份时，则对其进行解密。在本节中，我们将讨论可用的加密类型，然后查看如何配置 RMAN 以使用加密。

Oracle 提供了 3 种不同的加密模式：

- **透明模式**：透明模式加密不需要 DBA 干涉。为了使用该模式，必须配置 Oracle 加密电子皮夹(Encryption Wallet)。
- **口令模式**：口令模式加密需要在创建将要加密的备份或还原(在创建时)加密的备份时提供口令。在 RMAN 备份脚本中使用命令 set encryption on identified by password only 提供口令。这是本书中将采用的加密模式。
- **双重模式**：可以通过口令或 Oracle 加密电子皮夹来还原双重模式的备份。因为不需要安装 Oracle 加密电子皮夹，双重模式简化了备份的异地还原。可以使用 set encryption on identified by password 命令(注意缺少 only 关键字)创建双重模式加密备份。

使用 configure 命令可配置各种与 RMAN 备份加密相关的持久性设置。可以使用 RMAN configure 命令来设定以下相关的内容：

- 是否应该加密所有的数据库文件？
- 是否应该加密特定的表空间？
- 应该使用哪种可用的加密算法来加密备份？

如果正在使用基于 Oracle 加密电子皮夹的安全保护，则只需要设置 configure 命令所需的持久性 RMAN 设置。如果希望使用口令模式加密或双重模式加密，就需要使用 configure 命令配置持久性安全默认值，然后在启动备份时使用 set 命令设置备份口令，则在每次 RMAN 备份或恢复会话期间必须输入该口令。第 9 章将详细介绍 set 命令以及在备份期间它的用法。在下面的命令中，我们为整个数据库配置并启用备份加密。注意，如果没有配置 Oracle 加密电子皮夹，随后的任何备份都将会失败，除非使用 set 命令建立会话的加密口令(对此，我们省略了一些内容，但提供了使用 set 命令设置适当上下文的备份口令的示例)。

```
-- Configures default encryption.
   -- Uses transparent mode encryption by default.
```

```
RMAN> CONFIGURE ENCRYPTION FOR DATABASE ON;
-- For this session, we want password mode encryption,
-- so we have to set the
-- password. This is good only for this session, until we exit RMAN or issue
-- another connect command.
RMAN> SET ENCRYPTION ON IDENTIFIED BY robert ONLY;
-- Way ahead of ourselves, but this backs up the database!
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

如果满足下列情况，则使用加密备份归档的重做日志备份：

- 在进行归档重做日志的备份时执行 set encryption on 命令。
- 已经为整个数据库配置了加密，或者至少为数据库的一个表空间配置了加密。

也可以通过 configure 命令来确定所希望使用的加密算法。这些可用的算法可以在 V$RMAN_ENCRYPTION_ALGORITHMS 视图中找到，如下面的例子所示：

```
SQL> select algorithm_name from V$RMAN_ENCRYPTION_ALGORITHMS;
ALGORITHM_NAME
----------------------------------------------------------------
AES128
AES192
AES256
```

知道可用的加密算法后，现在就可以配置希望使用的默认加密算法，如下例所示：

```
RMAN> Configure encryption algorithm 'AES128';
 using target database control file instead of recovery catalog
new RMAN configuration parameters:
CONFIGURE ENCRYPTION ALGORITHM 'AES128';
new RMAN configuration parameters are successfully stored
```

8. 配置归档日志删除策略

可以配置 RMAN 来管理归档重做日志的删除策略。默认情况下，Oracle 将配置的备份保留策略应用于归档重做日志。在 Oracle Database 11g 中，还可以为归档重做日志另外配置自己的删除策略。在此环境下，配置的策略对 FRA 中以及存储在 FRA 之外的归档重做日志均有效。但是只有那些保存在 FRA 中的归档重做日志由 Oracle 删除。如果日志在 FRA 中，Oracle 将尽可能保存这些日志，除非有空间的需求才会删除。如果日志不在 FRA 中，则需要使用 delete obsolete 或 delete archivelog 命令才能将标记为过时的归档重做日志删除。下面的示例将使用 configure 命令配置归档日志删除策略，在这种情况下所有备份超过 3 次的归档重做日志将会被删除。

```
RMAN> Configure archivelog deletion policy to backed up 3 times to device
type disk;
new RMAN configuration parameters:
CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 3 TIMES TO DISK;
new RMAN configuration parameters are successfully stored
```

在 Oracle Database 11*g* 以前的版本中，归档重做日志删除策略仅对应用到备份数据库的归档重做日志有效。在 11*g* 以前的版本中，如果正在使用 FRA 和备用数据库，就可以配置 RMAN，在使用 configure archivelog deletion policy to applied on standby 命令将归档的重做日志应用于强制的备用数据库后将其标记为可删除。在这种情况中，一旦已经成功将归档的重做日志应用于强制的备用数据库位置，Oracle 就可以将其从 FRA 中删除。Oracle Database 11*g* 以及今后的版本保留了该功能。

3.5.4　使用共享服务器的情况

如果使用 Oracle 的共享服务器选项(在以前的 Oracle 版本中称为 MTS，或多线程服务器)，就必须配置一个 RMAN 使用的专用服务器，这是因为 RMAN 不能使用共享服务器会话连接到数据库。如果使用一个共享服务器体系结构，则可以参考 *Oracle Database Backup and Recovery Advanced Users Guide*(*11g Release 2*) 第 5 章，得到有关使用 Oracle Database 11*g* 共享服务器选项配置 RMAN 的更多信息。

最后，必须使用 SERVER=dedicated 语法来为服务器配置 Oracle Net 中的专用连接，如下面的示例所示(注意，Oracle Net 配置会经常改动，因此所需要的内容可能会有所不同)：

```
Rob1_ded =
    (DESCRIPTION=
      (ADDRESS=(PROTOCOL=tcp)(HOST=robpc)(port1521))
      (CONNECT_DATA=(SERVICE_NAME=rob1_ded)(SERVER=dedicated)))
```

3.6　RMAN 配置任务总结

本章介绍了大量信息，现在就概括一下在使用 RMAN 执行备份的数据库上应当执行的任务。下面是这些任务的总结：

(1) 确定是在 ARCHIVELOG 模式中还是在 NOARCHIVELOG 模式中运行数据库，并相应地配置数据库。大多数情况下，推荐 ARCHIVELOG 模式，因为它可以提供大量的恢复选项。

(2) 建议配置并应用 FRA。

(3) 设置 RMAN 使用的一个独立的数据库用户账户(不是 sys)。

(4) 在数据库参数文件中，将 CONTROL_FILE_RECORD_KEEP_TIME 参数设置为某个天数，这个数应大于等于要保留数据库备份的天数。保存是能够从 RMAN 得到的区域。应该确保 RMAN 的保留策略，CONTROL_FILE_RECORD_KEEP_TIME 参数，以及所有由磁带管理员(若采用磁带备份策略)建立的保留策略不产生冲突。

(5) 如果使用共享服务器，就要为 RMAN 设置一个连接数据库的专用服务器地址。

(6) 使用 RMAN 连接到目标数据库，确保数据库的设置正确(如果没有正确设置 RMAN 账户，就会显示错误消息)。

(7) 使用 configure 命令建立默认的 RMAN 值，特别要考虑下面这些配置情况：

- 配置磁带或磁盘备份的默认并行级别。设置一个默认值，这个默认值等于要创建备份的磁盘或磁带驱动器数。如果要使用多个磁盘驱动器来备份到 SAN，则应考虑使用并行信道来备份这些磁盘设备。
- 配置自动的信道和设备类型。配置与独立设备数目相同的信道。
- 配置自动化控制文件/数据库参数文件的自动备份操作。
- 如果拥有 ASO(Advanced Security Option，使用加密的许可选项)，则配置自动数据库备份加密。

(8) 有必要则配置保留策略。确认这个保留策略与其他保留策略(如与磁带管理系统关联的保留策略)同步。同样，如果有需要则为归档重做日志配置保留策略。

(9) 配置 RMAN，使其进行控制文件和 SPFILE 的自动备份。

(10) 使用 RMAN 执行产品数据库备份操作时，使用后面章节介绍的备份和恢复方法来测试 RMAN 配置。

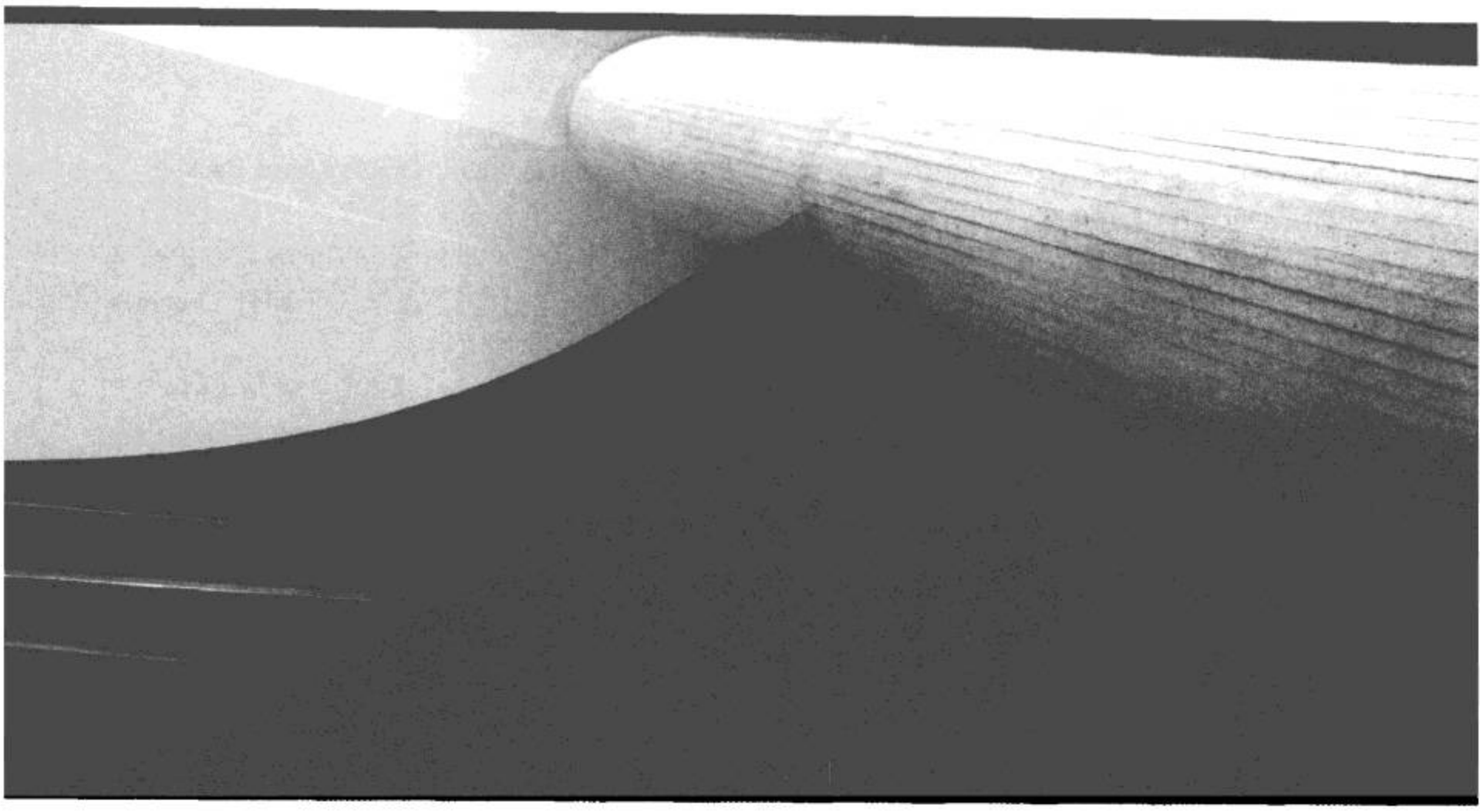
3.7 其他备份和恢复设置与配置问题

最后再介绍一些其他的数据库备份和恢复问题。在整个备份和恢复策略计划中，我们需要考虑到 RMAN 不会备份的一些文件，它们包括基本的 Oracle RDBMS 软件和参数文件(如 tnsnames.ora、names.ora、sqlnet.ora 等文件)。必须在整个备份与恢复计划中考虑到这些文件的备份与恢复。

还需要考虑针对 RMAN 备份和非 RMAN 备份的灾难计划。如何确保这些备份不受到洪水、火灾或地震的威胁？最好先做好准备，以应付突如其来的灾难。

3.8 小结

本章介绍了大量的基础知识，可以看到在开始使用 RMAN 前，必须先做一些前期的准备工作。首先，介绍了设置数据库为 ARCHIVELOG 模式的方法。接下来，介绍了 RMAN 命令行，并说明了如何配置数据库，包括设置口令文件和配置用户账户，以供 RMAN 使用。还介绍了配置 RMAN 默认设置，并强烈建议充分利用 RMAN 中的这一功能。本章还概述了 RMAN 配置任务，讨论了其他备份与恢复问题。



第4章

介质管理问题

Oracle Database 11*g* 中的 RMAN 实用程序主要关注的是将磁盘备份作为介质恢复解决方案的最佳方法。随着磁盘价格的下跌，海量存储区域网络(Storage Area Network，SAN)已经被许多数据中心使用。随着业务需求趋向于使用价格更加便宜而容量更大的磁盘，更好地利用可用存储空间成为 RMAN 功能(例如闪回恢复区)升级的重要依据。

RMAN 备份实用程序是按逻辑执行的过程，而写入到磁盘的操作正是 Oracle 数据库所擅长的工作。因此，当 Oracle 数据库开始改进其磁盘写入功能时，RDBMS 也会做相应的工作来改进性能。

但是，许多客户还没有实现无限磁盘存储。在许多情况下，数据库的大小及其位置都使得它无法备份到磁盘。或者，仍然存在要建立数据副本或异地归档的业务需求。因此，如果需要写入到老式的磁带设备，RMAN 需要做哪些工作呢？

Oracle 数据库的磁带备份需要第三方支持，这主要是因为市场上流行的连续介质子系统具有完全不同的性质。RMAN 开发人员通常使用那些大的软件供应商提供的可读写磁带产品，而不去尝试这些不同类型的磁带设备的不同系统调用。

Oracle 拥有自己的介质管理软件解决方案，称为 Oracle 安全备份(Oracle Secure Backup，OSB)。OSB 是完全集成的、RAMN 到磁带的解决方案，它不需要任何第三方供应商的软件插件。但是，自 10gR2 版本发布以来，OSB 仍然不成熟，虽然它价格低廉而很快在市场上获得成功。然而，在它完全成熟之前，大多数客户仍然会继续从一个拥有 Oracle RMAN 插件(后面将详细介绍该插件)的认证备份供应商处购买许可。

本章将介绍使用介质管理器直接在磁盘上备份数据库的体系结构概念，并从 RMAN 的角度，概括地介绍介质管理产品。我们将讨论 RMAN 端的设置，包括用磁带作为输出设备时的工作原理和变化情况。第 6 章到第 8 章详细介绍了市场上最流行的介质管理产品，并讨论这些产品的配置和使用。第 5 章概述了 Oracle 提供的介质管理新产品 OSB。

在使用 Oracle 10gR2 或 Oracle 11g 时，如果拥有单个磁带头且直接连接到希望备份的 Oracle 数据库的服务器上，则可以使用其 OSB 免费地实现将数据库备份到磁带的功能。换句话说，如果需要获得额外的备份到磁带的功能，则需要购买完整版本的 OSB，或从介质管理供应商处购买相关产品。

4.1 使用磁带备份的原因

Oracle 数据库的容量是一个关键因素。实际上在 10 年前，拥有几个 GB 的容量的数据库就可以称为大型数据库；现在大型数据库一般都拥有几个 TB 的数据容量。据报目前已经出现拥有 PB 级容量的数据库。数据库的平均容量可以达到 100GB，并且这一数字还在继续增长。因此，在执行备份操作时，即使企业中部署了 SAN，试图找到具有用于备份数据库的足够的连续磁盘空间也非常困难。

因此，数据库大小成为考虑使用磁带备份的首要理由。数据库的大小决定了是否需要使用磁带备份操作。即使磁盘价格在迅速下跌，购买大量硬盘需要的花费仍然巨大。从保存数据副本的角度(副本通常较少被用到)来看，磁带仍然相对便宜和可靠。当然，与磁带相比，磁盘备份的优点在于恢复速度更快，使用磁带方式，备份和还原相对更慢。与磁盘相比，磁带的价格仍然是使用磁带备份的一个引人注目的原因。

使用磁带备份的第二个理由是易管理性。通常，全企业备份策略的实现和执行是由中央系统上的人员集中管理的，这种方式使企业有能力购买大容量磁带盒式设备来存储来自许多数据源的数据。因此，不需要搬运分散在公司各处的磁带，不需要检查各个磁带设备的故障，也不需要训练人员学习新的软件，我们就可以分类和删除这些数据备份。

使用磁带备份的第三个经常忽视的理由是它们的可移植性。为了归档和避免灾难发生，可以很容易地将一堆磁带转移到异地，硬盘则无法方便地传输。

集中备份资源的缺点在于会导致其复杂化，尤其是 Oracle 数据库的复杂化。Oracle 数据文件、日志文件和控制文件的复杂性意味着我们不能简单地执行一个 OS 作业，让它在空闲时间复制文件。相反，我们必须使数据库做好执行备份操作的准备，然后通知

开始复制，最后重新配置数据库(与第 1 章介绍的方法大致相同)。

使用 RMAN 意味着避免了此类数据库的配置工作，任何时候、任何情况下都可以实施备份操作。不过，要将备份写入集中的磁带备份位置，就必须执行某些特殊的 RMAN 配置。

4.2　RMAN 和介质管理器概述

通过使用介质管理器，RMAN 将数据备份到磁带上。介质管理器(media manager)是由第三方软件商提供的软件，它将数据块中的数据流从 RMAN 信道进程传递到相应的磁带中。通常，介质管理服务器位于一个企业网络中。介质管理服务器(media management server)是一个集中式系统，在这里可以处理全企业的磁带备份操作。

计算机系统必须安装了相应地介质管理(简写为 MM)客户端软件才能使用介质管理器。介质管理客户端软件可以连接 MM 服务器，并且通过网络传送数据。如果 RMAN 要使用 MM 服务器，还需要另外的软件组件。安装客户端软件后，还必须为介质管理器安装 Oracle 模块。Oracle 模块(Oracle module)是一个用于 Oracle RDBMS 的软件插件，它可以连接 RMAN 与客户端介质管理软件，后者可以把数据传送到 MM 服务器。Oracle 的这个插件也称为介质管理库(Media Management Library，简写为 MML)。图 4-1 概述了使用介质管理器执行磁带备份操作的备份拓扑结构。

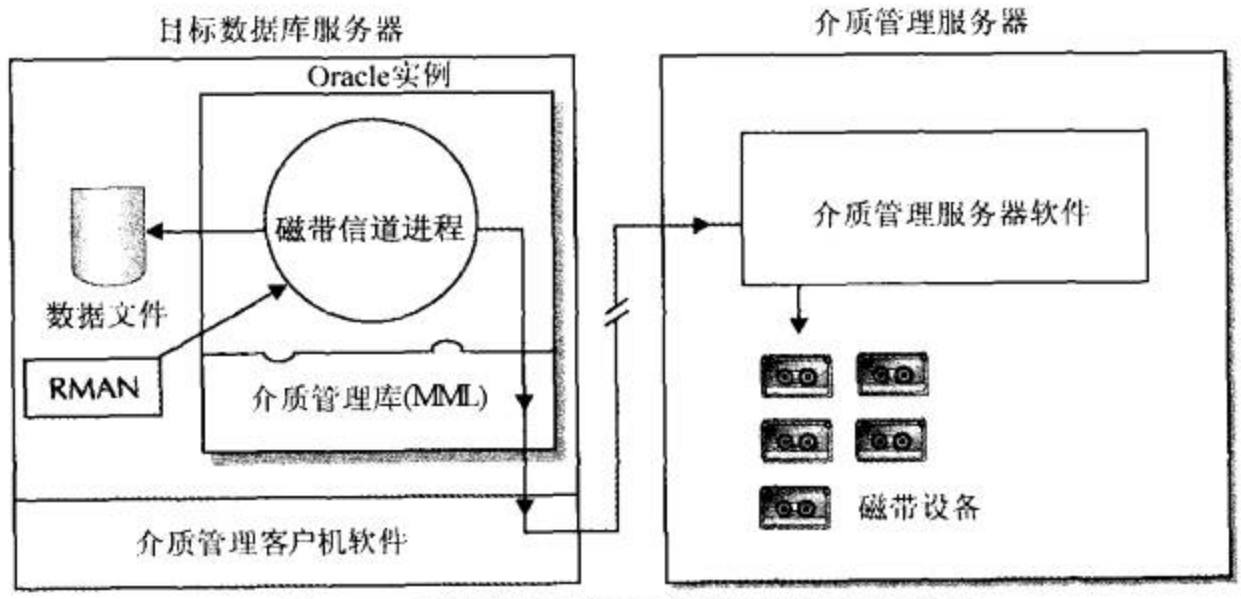


图 4-1　执行磁带备份操作时的网络拓扑结构

4.2.1　介质管理器目录

介质管理器是我们使用的整个备份系统中的一个独立子系统。前面介绍过，介质管理器具有 3 个基本的组件：与 Oracle 结合的介质管理库(MML)，介质管理客户端以及介质管理服务器。MM 服务器具有多种组件，这些组件的规范是由供应商提供的。不过，所有的 MM 服务器都必须具有一些相同的组件，其中就本章讨论的范围来看最重要的组件是介质管理器目录。

介质管理器目录是 MM 服务器上的信息数据库，它保存了关于物理磁带本身、磁带

的访问权限和磁带存储内容的相关信息。备份完成时，介质管理器目录则记录 RMAN 文件句柄，句柄(handle)是使用 RMAN 执行备份操作时创建的备份片名称。如果执行磁盘备份操作，句柄就是物理文件名。如果执行磁带备份操作，介质管理器目录中使用的句柄就是备份所在的磁带位置。

RMAN 完成磁带备份操作时会向介质管理器提供句柄名，介质管理器把该句柄记录在目录中。需要执行还原操作时，RMAN 会基于自身的目录向介质管理器请求指定的句柄。随后，介质管理器查找这个句柄，将该句柄与指定磁带关联在一起，同时判断磁带是否有效。如果磁带有效，介质管理器会使用这个磁带，并且开始将数据流传送回 RMAN，这样就可以重新构建数据文件。

4.2.2 介质管理器的其他软件组件

除了介质管理器目录以外，MM 服务器还包含两个基本组件：

- 设备代理程序(device agent)：设备代理程序是负责使用实际磁带设备并在该设备上传送数据的组件。
- 自动机器接口(robotic interface)：自动机器接口指控制所有自动机器的软件，这些自动机器负责在磁带满时或在请求写入已填满磁带时更换磁带。

Oracle 中的 RMAN 不了解这些组件的存在。RMAN 只是简单地把命令请求发送给 MML，随后 MM 软件会处理相应的所有事件。不过，由于备份和恢复的成功取决于这些组件，所以熟悉这些软件组件是至关重要的。使用 RMAN 出现的问题大多数都与设备代理程序或自动机器接口有关，但从 RMAN 接口几乎觉察不到这些问题。

4.2.3 介质管理库(MML)

MML 是一个库文件，它将 RMAN 的一致性的备份或还原请求解释为在介质管理服务器上的具体的系统调用，以实现该要求的操作。MML 与 MM 客户端软件和 MM 服务器软件一样是由同一个供应商提供的，但是需要单独购买 MML 和取得许可。

首次分配磁带信道时，MML 作为一个集成的库文件加载到 Oracle 内存空间，它是 Oracle RDBMS 软件的逻辑部分，因此 RMAN 可以生成正确的 MM 客户端软件调用。这个集成实际上非常简单：分配磁带信道时，Oracle 加载一个名为 libobk.so 的文件。这个文件位于 ORACLE_HOME/bin 目录中，它只是一个到要使用的 MML 文件的符号链接。在 Windows 平台上，Oracle 会在搜索路径中查找一个名为 orasbt.dll 的文件。不论是哪一种介质管理器，其介质管理 DLL 文件都名为 orasbt.dll，介质管理器通常将其写入到 WINDOWS\system32 目录中。如果这个文件没有位于该目录中，在系统路径环境变量中会有一个能够查找到 orasbt.dll 文件的搜索路径。

在接下来的 4 章中将讨论链接进程，我们可以通过链接进程在分配信道时将供应商的 MML 文件建立为一个 RMAN 启动。出于测试目的，Oracle 提供了一个测试 MML 文件，这个库文件允许用户分配磁带信道，但将备份数据写入磁盘。下面的 RMAN 实习将介绍使用测试 MML 文件的方法。

RMAN 实习：使用 Oracle 默认 SBT 接口测试磁带信道

实习注意事项

需要访问一个足够大的磁盘空间，并在磁盘上创建一个目录来放入备份片。在这个示例中，使用加载点 u04，创建名为 backup 的目录。如第 2 章所述，需确认有足够的空间用于备份操作，并留意到备份位置的磁盘 I/O 操作。另外，还要尝试在不同于放置实际数据库的另外一个控制器上分配空间。

步骤 1　创建备份目录：

```
$>cd /u04
mkdir backup
```

步骤 2　确认建立权限，以便 Oracle 数据库(像在用户操纵下安装软件一样)可以在备份位置上写入数据：

```
ls -l backup
```

步骤 3　启动 RMAN 并连接目标数据库。在下面的示例中，在本地连接目标数据库 PROD，此时 env 命令显示 ORACLE_SID=PROD:

```
$>Rman
Rman> connect target
```

步骤 4　运行备份操作，在信道分配期间使用 PARMS 参数指定 Oracle 测试库文件。还需要指定 RMAN 写入备份的位置 BACKUP_DIR 目录，如下所示，这个位置为/u04/backup:

```
<RMAN> run {
2> allocate channel x1 type 'sbt_tape'
3> PARMS="SBT_LIBRARY=oracle.disksbt,
4> ENV=(BACKUP_DIR=/u04/backup)";
5> backup datafile 1 format='%U';}
```

作为选择，可以使用永久性配置命令设置 Oracle 库(但是需要记住，已经完成了 Oracle 库的设置，不要让其空闲太长时间)：

```
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'SBT_LIBRARY=oracle.disksbt,ENV=(BACKUP_DIR=/u04/backup)';
```

上面这些命令的输出如下所示：

```
allocated channel: x1
channel x1: SID=27 device type=SBT_TAPE
channel x1: WARNING: Oracle Test Disk API
Starting backup at 25-SEP-09
channel x1: starting full datafile backup set
channel x1: specifying datafile(s) in backup set
```

```
input datafile file number=00001
name=/home/oracle/app/oracle/oradata/v112/
system01.dbf
channel x1: starting piece 1 at 25-SEP-09
channel x1: finished piece 1 at 25-SEP-09
piece handle=04kq4cbe_1_1 tag=TAG20090925T102902 comment=API Version
2.0,MMS Version 8.1.3.0
channel x1: backup set complete, elapsed time: 00:02:07
channel x1: starting full datafile backup set
channel x1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel x1: starting piece 1 at 25-SEP-09
channel x1: finished piece 1 at 25-SEP-09
piece handle=05kq4cfd_1_1 tag=TAG20090925T102902 comment=API Version
2.0,MMS Version 8.1.3.0
channel x1: backup set complete, elapsed time: 00:00:02
Finished backup at 25-SEP-09
released channel: x1
```

如果试图分析介质管理器备份解决方案可能存在的问题或无法使备份工作时，那么上面的实习是一个非常有用的测试。通过分配一个“伪造的”磁带信道，我们可以确保RMAN的配置是正确的。

警告：
不要在产品备份时使用测试MML文件。如果要在产品环境中执行磁盘备份操作，就分配磁盘信道。伪造MML的性能非常糟糕，这是因为RMAN为磁带而不是磁盘分配内存缓冲区，因而磁盘写操作的速度远大于磁带写的速度这一点未得到体现。

如果没有成功地加载供应商的MML文件，并且在信道分配的PARMS部分中没有指定要使用的Oracle磁盘SBT接口，试图分配磁带信道时就会得到一条错误消息：

```
RMAN-00571: ===========================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===
RMAN-00571: ===========================================
RMAN-03009: failure of allocate command on x channel
     at 12/05/2005 21:26:43
ORA-19554: error allocating device, device type: SBT_TAPE, device name
ORA-27211: Failed to load Media Management Library
```

4.2.4 连接MML的接口

链接Oracle与MML时，意味着RMAN可以通过它来传递命令以连接MML，或者进一步说，在数据库服务器上安装MM客户端软件。但是，我们怎么才能知道使用的是

哪一个 MM 服务器呢？

要指定 MM 服务器，必须在 RMAN 会话中传递指定服务器名的环境参数。具体来讲，在分配磁带信道时，将服务器名指定为环境变量。在前面的 RMAN 实习示例中，我们使用 allocate channel 命令的 PARMS 选项来传递环境变量。不同的 MM 产品具有不同的环境变量。如下所示，VERITAS NetBackup 要求使用 NB_ORA_SERV 参数：

```
Allocate channel t1 type 'sbt_tape'
    PARMS="ENV=(NB_ORA_SERV=storage1)";
```

在上面的示例中，MM 服务器的名称为 storage1，同时数据库服务器在 MM 服务器中已注册并具有写磁带设备的权限。

除了服务器名之外，还可以在信道分配时传递其他一些参数来利用服务器上的管理功能。例如，NetBackup 允许我们指定用于备份操作的类或调度，而 EMC Networker 允许我们指定资源池。第 6、7 章将详细介绍这些参数。

4.3　SBT API

RMAN 可以不受影响地使用不同的介质管理器，这是因为不管加载了哪一种 MML，RMAN 都会发送相同的命令。Oracle 使用 SBT API 来开发 RMAN，SBT API 是一种通用 API，提供给 Oracle 数据库备份操作编码集成产品的第三方软件商，是 RMAN 向介质管理器发送命令的工具。

SBT API 负责向 MM 服务器发送命令以初始化在磁带上备份文件的创建，还发送基于介质管理器目录中的文件句柄来查找先前备份的命令。SBT API 可以发送删除备份的命令，也可以写新备份以及从备份位置读取备份。Oracle RMAN SBT API 有 1.1 和 2.0 两种版本。1.1 版本是与 Oracle 8.0.x 一起发布和使用的。其后，RMAN 使用 2.0 版本的规范生成介质管理器调用，运行备份时可以在 RMAN 输出中看到这个版本号：

```
channel x1: finished piece 1 at 25-SEP-09
    piece handle=05kq4cfd_1_1 tag=TAG20090925T102902 comment=API Version
    2.0,MMS Version 8.1.3.0
    channel x1: backup set complete, elapsed time: 00:00:02
    Finished backup at 25-SEP-09
```

RMAN 还会返回信道分配时初始化的 MML 版本，这可以在信道分配期间的 RMAN 输出中看到：

```
allocated channel: x
    channel x: sid=12 devtype=SBT_TAPE
    channel x: VERITAS NetBackup for Oracle8 - Release 3.4GA (030800)
```

上面的方法不仅可以确定 MML 版本，而且可以说明已经成功地链接了 MML 与 RMAN，否则 RMAN 输出中就不能析取出版本信息。

4.4　磁带备份的全过程

本节将讨论磁带备份的全过程，说明对 SBT API 生成的不同调用，以及介质管理器处理这些调用的方法。再次需要注意的是，我们所介绍的是一般概述，而具体细节是由编写集成 MML 的供应商来处理的。

分配磁带信道时，RMAN 会在目标数据库上生成一个服务器进程，这个服务器进程随后生成 sbtinit()的 SBT API 调用，该调用初始化 MML 文件并将这个文件加载到内存，同时还向 RMAN 返回 MML 支持的 SBT API 版本。调用 sbtinit()后，RMAN 还会调用 sbtinit2()，sbtinit2()调用将向介质管理器软件提供更详细的配置信息。

RMAN 分析备份命令后会执行远程过程调用(RPC)，RPC 会生成对 sys.dbms_backup_restore.backuppiececreate 的调用。这时，信道进程将调用 sbtbackup()，该调用在指定的磁带位置处理备份片的创建。sbtbackup()调用会通知介质管理器 Oracle 将开始推入数据块流，这样介质管理器会为数据流准备好磁带设备。

RMAN 输入缓冲区被填满后会执行内存对内存的写操作把内容写入输出缓冲区。输出缓冲区被填满时，信道进程调用 sbtwrite2()，从而将输出缓冲区的内容写到磁带上(参阅第 2 章输入缓冲区的内容)。一般情况下，这种方式会在 MM 服务器上使用设备代理程序来访问磁带自身。

当用于特定备份集的所有输出缓冲区被清空，并且 sbtwrite2()调用的工作也已完成时，信道会话会调用 sbtclose2()，该调用将彻底清空所有介质管理器缓冲区，并且将备份片提交给磁带。

完成备份片后，信道进程调用 sbtinfo2()，该调用确认介质管理器目录已记录了备份片。sbtinfo2()调用向介质管理器目录请求备份的磁带、磁带位置和备份的完整时间，然后将备份片句柄写入介质管理器目录。

在确认备份片位置后，信道进程调用 sbtend()，该调用清除剩余的资源，并将它们释放，以用于其他数据库操作。最后执行的是收回信道进程的动作，它在目标数据库中完成。

4.5　磁带还原的全过程

我们迟早会测试生成的所有备份，此外还会执行还原操作。在还原操作期间，SBT API 将经过一系列步骤还原磁带备份到数据库中。本节将简要介绍还原操作期间的 SBT API 过程。

为还原分配磁带信道时，RMAN 会在目标数据库上创建一个服务器进程。随后，信道进程调用 sbtinit()来初始化介质管理器软件。这一步骤与备份操作的初始化相同：MML 文件被加载到内存中。

基于 RMAN 中 restore 命令的参数，RMAN 会检查 RMAN 目录来确定需要进行还原的备份句柄名。随后，它使用 sbtrestore()调用将请求的备份片句柄传递给介质管理器，该调用告诉介质管理器要为还原操作准备好相应的磁带，即使用介质管理器目录来查找相

应的磁带，如果有必要则向自动机器指令集传递获取磁带的命令。加载磁带后，磁带必须绕回到备份片的起始点。

为还原准备好磁带后，信道进程调用 sbtread2()函数从磁带设备读取数据并将数据流发送至 Oracle 进程。这些数据被加载到输入缓冲区，然后被写入输出缓冲区，最后被写入控制文件指定的数据文件位置。

在磁带上检测到备份片的结束点时，磁带信道进程调用 sbtclose()函数来断开该备份片所在的磁带，这表明 Oracle 结束了对这个磁带的操作。如果执行还原操作时需要多个备份片，信道进程会返回第二个步骤并为另一个备份片调用 sbtrestore()。

还原操作完成并且 RMAN 不再请求备份片时，信道进程调用 sbtend()函数，该函数清理并释放信道资源以供其他方面使用。然后信道进程结束，此后介质管理器就可以卸载已经请求过的磁带。

4.6　使用 sbttest 和 loadsbt.exe

前面曾经提到过，通常总是有一些指示说明是否成功链接了 MML 与 Oracle，例如，来自信道分配的信息说明了 MML 的版本。不过，这些指示不能保证最后的成功，在更下层次的网络拓扑中(MM 客户端级或 MM 服务器)仍然可能出现故障。Oracle 提供了一个名为 sbttest 的实用程序，可用它来测试以确保 RMAN 能够使用 MM 配置来执行磁带备份操作。从命令行可以调用这个实用程序，该程序执行一个完整的测试：sbttest 会向磁带写一个数据块，然后请求读取这个数据块。通过这种方式，sbttest 运行备份操作期间执行的所有的 SBT API 函数，并确认这些函数能否成功。

sbttest 的用法非常简单：只需从命令提示符运行。确认完成了所有的 MM 配置，然后进入 RMAN 运行环境中的命令提示符，并输入 sbttest 和测试文件名。下面的代码把 4.5 节(磁带还原的全过程)中列出的每一个 sbt()调用执行一遍，并在输出中显示每个调用是否成功：

```
[redacted]/u02/home/usupport> sbttest oratest_061902
    The sbt function pointers are loaded from libobk.so library.
    NetWorker: Cannot contact nsrexecd service on horatio.hadba.com,
    Service not available.-- sbtinit succeeded
    NetWorker: Cannot contact nsrexecd service on horatio.hadba.com,
    Service not available.-- sbtinit (2nd time) succeeded
    sbtinit: Media manager supports SBT API version 2.0
    sbtinit: vendor description string=NMO v3.5.0.1
    sbtinit: allocated sbt context area of 536 bytes
    sbtinit: Media manager is version 3.5.0.1
    sbtinit: proxy copy is supported
    sbtinit: maximum concurrent proxy copy files is 0
    -- sbtinit2 succeeded
    -- regular_backup_restore starts ..............................
    MMAPI error from sbtbackup: 7501, nwora_index_ssinfo:
    index connect to cervantes.windba.com failed for client
```

```
horatio.hadba.com: Program not registered
-- sbtbackup failed
```

从最开始简单地使用二进制方法来指示成功或失败，如今的 sbttest 实用程序已经相当完善。现在，我们可以传递许多参数来测试 MM 系统的所有过程，包括命名要测试的数据库，修改 sbttest 写入的数据块数目，以及处理 sbttest 写入磁带的文件。从命令提示符简单地输入 sbttest 会给出所有可以使用的参数开关和简单的文本说明。

sbttest 实用程序只适用于 Unix 平台；在 Windows 平台上，可以向 Oracle Support 请求 loadsbt.exe 实用程序。但是，loadsbt.exe 实用程序不具有与 sbttest 相同的性能，它只是简单地在搜索路径中查找 orasbt.dll 文件。如果查找到该文件，loadsbt.exe 实用程序会试图加载该文件，其加载方法与 Oracle 在磁带备份操作期间加载的方法相同。如果可以加载 orasbt.dll 文件，loadsbt.exe 实用程序会通知用户，但它不会在磁带上写入数据块，所以我们无法查看整个 MM 配置工作。因此，loadsbt.exe 的作用不如 sbttest。

4.7　介质管理错误

RMAN 中的错误报告在报告 MM 问题时，其方式与报告其他问题的相同，这样容易引起混淆。要知道，检查 RMAN 错误时定位错误来源(RMAN、目标数据库、恢复目录数据库或介质管理器)是非常重要的。

我们有具体的方式可以判断返回到 RMAN 的错误是否与介质管理器有关。如果没有正确链接 MML，某些错误是显而易见的，前面已经介绍过这类示例。然而，如果正确链接了 MML 与 Oracle 软件，怎样才能判断错误是否与 MML 有关呢？

错误的种类相当多，不过与 MML 有关的最常见错误是 ORA-19511。这种错误实际上是一种空格错误，即 Oracle 没有提供文本，而是提供了被 MM 视为错误的内容。因此，如果出现下面这种错误，无疑已经正确地链接了 MML，并且遇到的这个问题与介质管理器有关：

```
RMAN-19511: sbtbackup: Failed to process backup file
```

其他 MM 问题的指示并不明确，不过可以区分错误的来源。例如，如果错误堆栈中看到“sequential file”，则说明我们就是在处理磁带备份，并且问题是由读或写磁带上的序列文件失败而引起的。下面是另一个常见错误 ORA-27206：

```
RMAN-10035: exception raised in RPC: ORA-27206: requested file
    not found in media management catalog
```

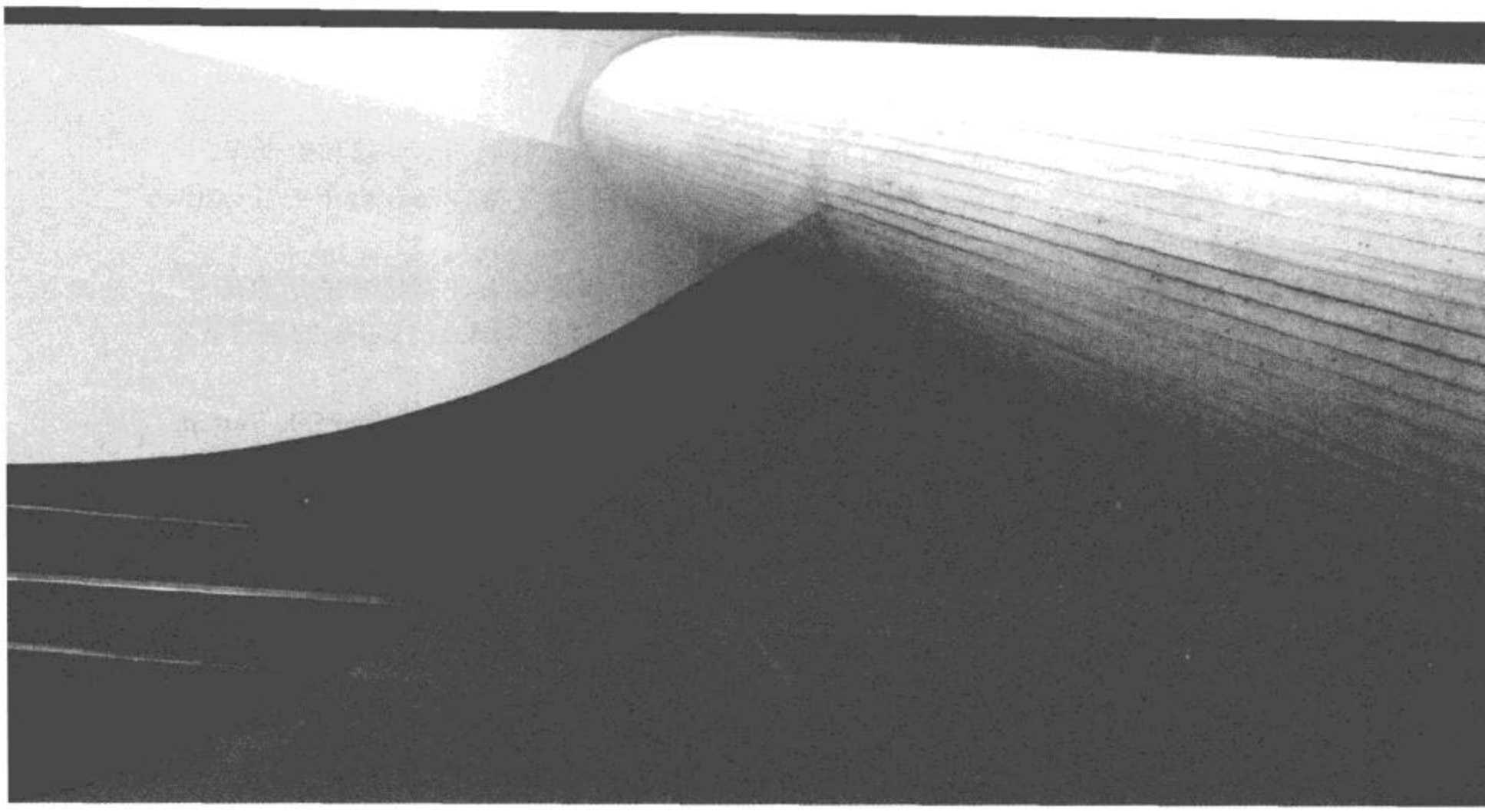
此外，上面的文本说明问题与 MM 目录有关，我们应当在这个 MM 目录中查找和解决问题。

除了错误以外，我们在 RMAN 还可能遇到的暂停通常与 MM 问题有关。例如，RMAN 生成对介质管理器的 sbwrite()调用时，RMAN 不大可能知道完成该调用的时间，因此 RMAN 不提供操作的超时时间，而是无限期地等待介质管理器返回一个成功的写操作或返回一

个错误。如果介质管理器在等待一个没有超时的特定事件(如磁带切换或磁带装入)，RMAN 也会等待。前面说过，RMAN 没有超时。因此，如果发现 RMAN 完成操作所花的时间相当长，并且在 V$SESSION_LONGOPS 视图中看不到任何进展(参阅第 2 章)，就应当首先检查介质管理器的未捕获错误或事件(如磁带装入或磁带切换)。

4.8　小结

本章讨论了 RMAN 利用第三方软件商的介质管理软件在磁带上生成备份的方法，介绍了 RMAN 使用 SBT API 来执行磁带备份和还原操作的全过程，并简要讨论了 RMAN 中的介质管理错误



第5章

Oracle 安全备份

Oracle 安全备份(Oracle Secure Backup，OSB)是一种可靠、全面、集中式的磁带备份管理解决方案。OSB 为分布式混合平台环境下的异构数据提供保护。用于保护诸如 Oracle Home 目录下的 Oracle 数据库和文件系统的数据。OSB 不仅提供磁带备份，通过 Oracle 安全备份云模块(Oracle Secure Backup Cloud Module)也可以将集成的 Oracle 数据备份发布到第三方云(网络)存储中。OSB 提供两类磁带管理版本：OSB 及 OSB Express。

本章主要讨论 OSB 的特性、接口、OSB 的组件，以及安装和配置 OSB 的过程。

5.1 OSB 的特性

- 支持 SAN、Gigabit 以太网和 SCSI 环境中所有主要的磁带驱动器和磁带库。
- 支持网络协议 v4(IPv4)，网络协议 v6(IPv6)以及所有支持 IPv6 平台中的 IPv4/IPv6 混合应用环境。
- 支持访问所有本地及远程文件系统，支持访问网络中任意位置的所有未使用(Network File System，NFS)及(Common Internet File System，CIFS)的磁带设备。
- 支持光纤附属设备。
- 从运行在 Linux 和 Microsoft Windows 操作系统上的(Oracle Cluster File System，OCFS)备份及还原数据。
- 允许使用通配符和扩展名列表来指定备份的文件。
- 执行多级增量备份、双路数据库备份、跨越多个卷的备份。
- 提供自动磁带驱动设备共享，减少磁带写保护造成的空闲。
- 使用直接指向块的定位和直接访问还原来避免读取磁带块以定位备份。
- 为 Oracle 9*i* 及以前版本和文件系统的备份提供高性能可配置加密。
- 从 11.1.0.6 版本开始支持快速备份压缩。
- 支持 ACSLS(Automated Cartridge System Library Software，自动录音带盒系统库软件)及跳接(Vaulting)。
- 在备份及还原期间报告进展情况。
- 与 RMAN 紧密集成。
- 在 Oracle Database 10g Release 2 以后，支持与 Oracle 企业管理数据库(Enterprise Manager Database) 及网格控件(Grid Control)的集成。
- 允许使用 Oracle 安全备份云模块(Oracle Secure Backup Cloud Module)将数据直接备份到云存储中。
- 涉及恢复管理器和介质管理器的问题的单一联系点。
- 包含 Oracle Secure Backup Express，可方便应用于 Oracle 数据库，方便应用于将服务器备份及还原到单一磁带设备。

5.1.1 OSB 和恢复管理器

OSB 是应用于 RMAN 上的介质管理层。OSB 提供了 SBT 接口，RMAN 可利用该接口将数据库文件备份到磁带。OSB SBT 是唯一支持 RMAN 加密备份并将未使用块直接压缩到磁带的接口。RMAN 消除提交的 undo 备份的能力只对 OSB 有效，对其他介质管理器无效。Oracle Database 11g 通过为 SBT(系统备份到磁带)和磁带建立共享缓冲，从而消除了从 SBT 到磁带缓冲的复制过程，进而减少了 CPU 开销。与其他介质管理器比较，OSB 能够更好地与 Oracle 企业管理器集成(OEM)。使用 OEM 管理磁带，介质服务器和磁带设备专用于 OSB。

5.1.2　OSB 和 OSB Express 的区别

下面是 Oracle Secure Backup 和 Oracle Secure Backup Express 共有的特性：

- 与 RMAN 集成，用于 Oracle 数据库的联机磁带备份和还原
- 文件系统数据的备份和还原
- 与 Oracle 企业管理器集成(10gR2 和更高版本)

下面是 Oracle Secure Backup 独有的特性：

- 备份到 Real Application Clusters (RAC，实时应用集群)环境
- 与 Oracle 企业管理器的网格控件集成(Oracle Database 10gR2 和更高版本)
- 在备份环境中支持多个磁带驱动器使用
- 支持光纤附属设备
- 磁带的数据库备份加密
- Oracle 快速备份压缩(Oracle Database 11gR1 和更高版本)
- 支持 ACSLS 及跳接
- 分布式服务器与/或磁带设备的网络化备份的特性

5.1.3　备份加密

RMAN 和文件系统备份操作可使用 OSB 加密技术。数据在通过网络传输，或者被写到本地附属磁带设备前在服务器端加密。当 RMAN 将数据从 SBT 传递到 OSB 后，则数据库数据被加密。如果来自 SBT 的 RMAN 数据已经被加密，则不再执行加密操作。对于加密的磁盘备份，需要特殊的 Oracle 数据库高级安全选项(Advanced Security Option，ASO)的许可。

5.1.4　快速数据库备份压缩

快速数据库备份压缩通常仅在购买(Oracle Advanced Compression Option，ACO)后才可以使用。

5.1.5　Oracle 安全备份云模块

Oracle 安全备份云模块独立于 OSB 磁带管理版本。该模块目前仅支持 Amazon S3 (Simple Storage Service)，未来可能会支持其他服务供应商提供的云存储服务。Oracle 安全备份云模块的许可数量取决于备份到云模块的 RMAN 信道数量，而与 Oracle 数据库备份的数量无关。例如，若拥有 4 个 OSB 云模块许可，则可用于备份两个 Oracle 数据库，其中每个数据库使用 2 个 RMAN 信道，或者支持 1 个使用 4 个 RMAN 信道的 Oracle 数据库的备份。

5.2　OSB 的接口

图 5-1 展示了可用来访问 OSB 的接口，具体描述如下。

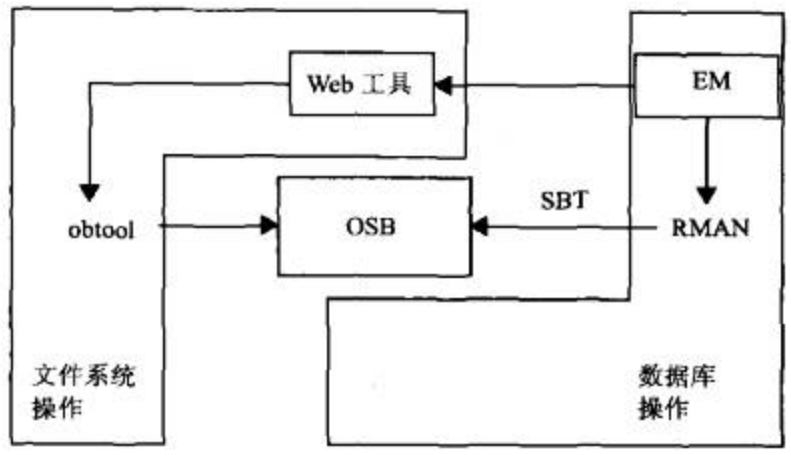


图 5-1　OSB 接口

- **Oracle 企业管理器数据库控件和网格控件**：作为一组基于 GUI 管理 VSB 的工具，大多数 OSB 任务可以通过 OEM 执行。OSB 管理服务器可以配置为 OEM 网格控件的目标服务器，可以用来执行文件系统备份和还原操作。
- **OSB Web 工具**：该接口是基于浏览器的 GUI，通过该 GUI 可配置管理域，浏览备份目录，以及管理文件系统数据的备份和还原，还可以执行 OEM 中不可以执行的其他任务。Web 工具使用运行在管理服务器上的 Apache Web 服务器。
- **OSB 的命令行接口(obtool)**：该命令行程序是 OSB 的主要接口位于 OSB 主目录的 bin 子目录中。使用 obtool，可以登录到管理域以备份和还原文件系统数据以及执行配置和管理任务。可以在管理域中的任何主机上运行 obtool。
- **恢复管理器命令行接口**：RMAN 命令行接口可用于配置和初始化数据库备份与恢复操作以使用 OSB。RMAN 实用程序位于 ORACLE_HOME 目录下的 bin 子目录中。只要客户端能够连接到目标数据库，RMAN 命令行客户端可以在任何数据库主机上运行。为了使 RMAN 使用 OSB 建立备份，OSB SBT 必须与目标数据库运行在同一主机上。

5.3　OSB 的组件

管理域是一个备份和还原操作的管理单元，它包含一组主机。当配置 OSB 时，角色将分配给域中的每个主机。一个主机可能由如下的一个或多个角色组成：

- **管理服务器**：启动和监控备份和恢复作业是由运行在管理域中的管理服务器完成的。管理服务器可能也运行除了 OSB 之外的其他应用程序。
- **介质服务器**：驻留辅助存储设备，例如磁带驱动器或磁带库。每个管理域至少定义一个介质服务器。
- **客户端**：本地数据由 OSB 备份的主机。在每个管理域中将定义一个或多个客户端。管理域中的大多数主机都是客户端。

图 5-2 显示了一个 OSB 管理域。该域包括一个管理服务器、一个带有附属磁带库的介质服务器、3 个客户端以及 5 个主机。

图 5-3 显示了一个包含单个 Linux 主机的 OSB 管理域。这个 Linux 主机假定同时扮

演管理服务器、介质服务器和客户端的角色。为这个 Linux 主机配置了一个具有本地附属磁带库的 Oracle 数据库。注意，该配置就是稍后将介绍的安装示例的配置。

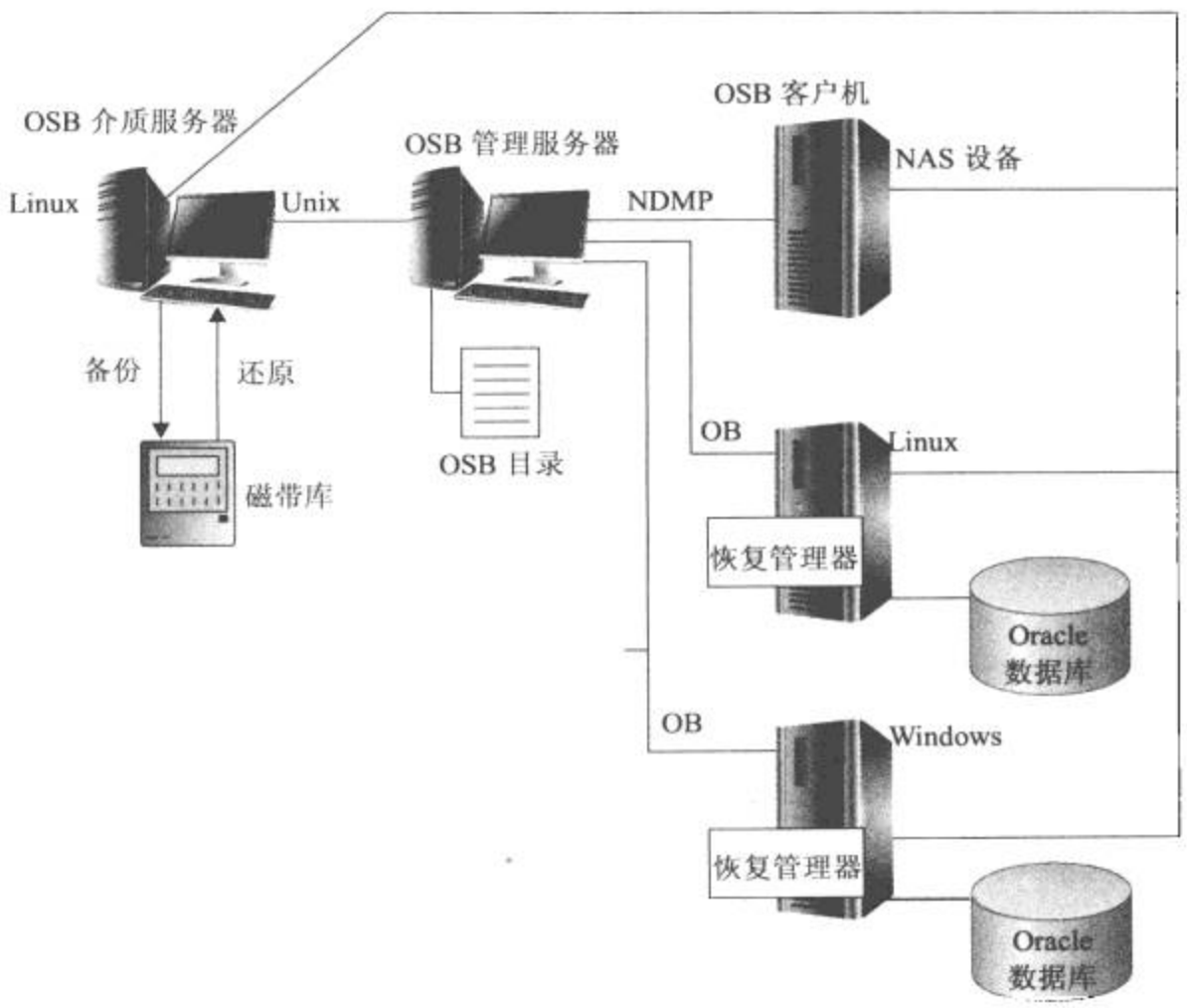


图 5-2　OSB 管理域

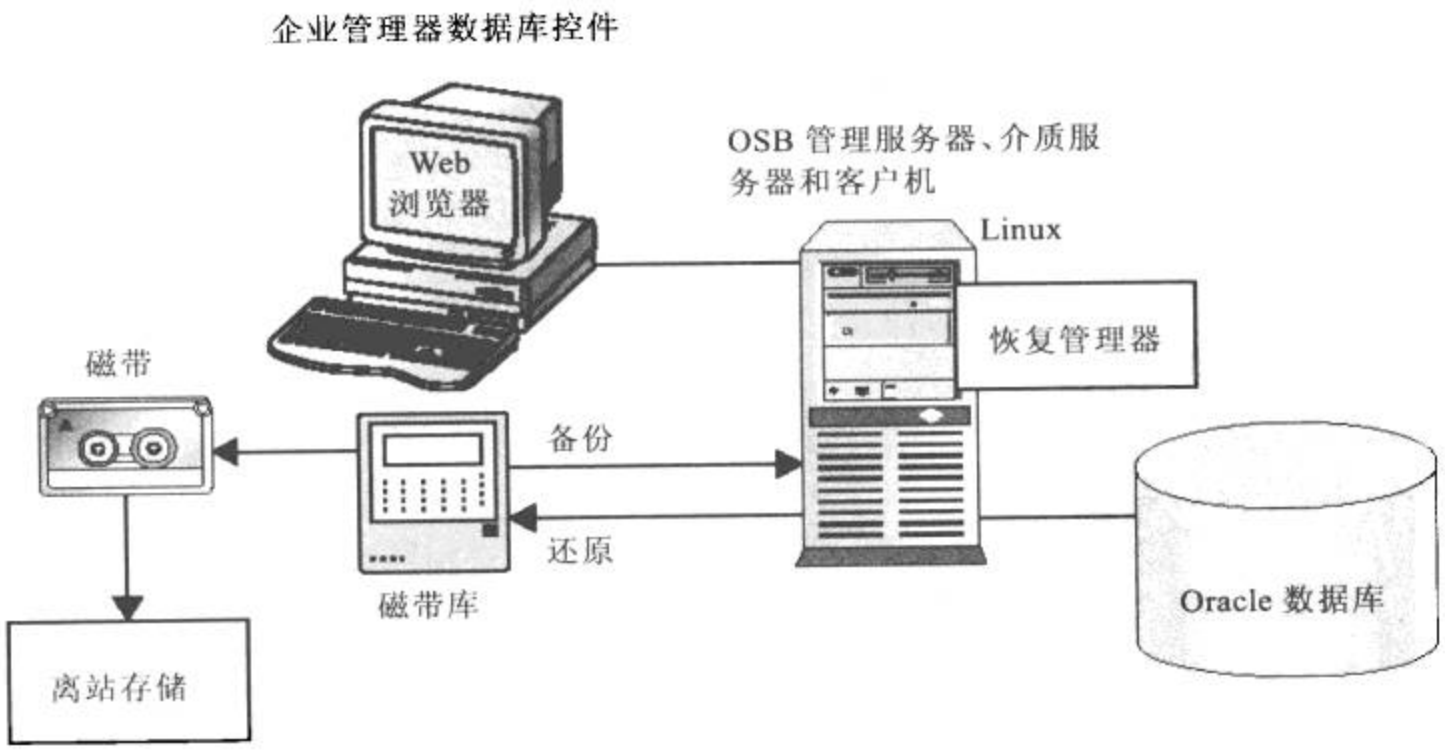


图 5-3　只有一个主机的 OSB 管理域

5.3.1 OSB 守护进程(Daemons)

管理域使用如下 7 类 OSB 守护进程：

- **Service 守护进程**：该守护进程在管理服务器、介质服务器和客户端之上运行。通过该守护进程可以访问管理服务器上的 OSB 配置数据。Service 守护进程也可以按照 Schedule 守护进程的请求运行相应任务。在介质服务器或客户端上，Service 守护进程负责处理管理域上的成员关系。
- **Schedule 守护进程**：该守护进程只在管理服务器上运行，作为 OSB 的调度器。
- **Index 守护进程**：该守护进程只运行在管理服务器上管理备份目录。当备份完成或目录被还原或浏览操作被访问时启动。
- **Apache web server 守护进程**：该守护进程只在管理服务器上运行，用于提供 Web 工具接口。
- **NDMP 守护进程**：该守护进程在介质服务器和客户端上运行，为它们之间提供数据通信。
- **Robot 守护进程**：该守护进程在介质服务器上运行，处理磁带库中的磁带。当需要操作磁带时，Service 守护进程将为每个磁带库启动一个 Robot 守护进程。
- **Proxy 守护进程**：该守护进程在客户端上运行，用于验证用户对 SBT 的备份和还原操作的访问。

5.3.2 主机访问模式

可以通过两种访问模式与管理域中的主机通信：

- **Primary**：对于主站访问模式，在主机上安装 OSB。OSB 守护进程运行该模式。Oracle 数据库一般位于通过该模式访问的主机上。在 OEM 中将该模式归类为“native”访问模式。在 OSB web 工具上，该模式称为“OB”访问模式。
- **NDMP**：网络数据管理协议(Network Data Management Protocol，NDMP)主机是由第三方供应商(例如 DinoStor、Mirapoint 和 Network Appliance)提供的存储设备。使用供应商特有的实现，NDMP 主机使用 NDMP 协议备份和还原文件系统。虽然 OSB 软件没有安装在 NDMP 主机上，但可以通过 NDMP 访问 OSB。

5.3.3 管理数据

OSB 将管理域的信息安排为管理服务器上 OSB 主目录中的文件层次。OSB 主目录是安装 OSB 的目录。

图 5-4 显示了 OSB 主目录的目录结构。所有平台具有相同的目录结构，但是 Linux 和 Unix 系统的默认主目录为/usr/local/oracle/backup，而 Microsoft Windows 系统的主目录是 C:\Program Files\Oracle\Backup。

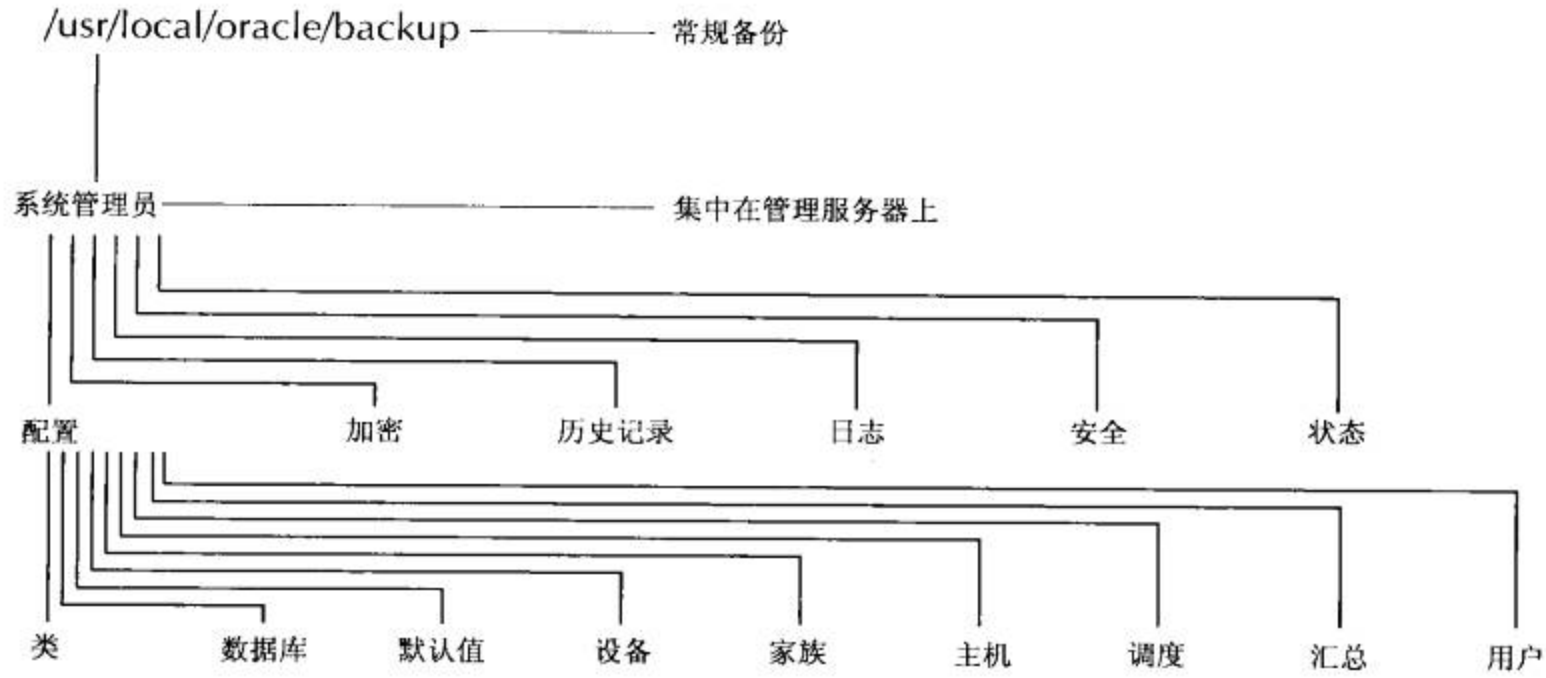


图 5-4　管理服务器的目录

域范围的实体(例如介质家族、类和设备)均包括在管理数据中。图 5-4 显示了配置目录如何包含一些子目录。这些子目录代表由 OSB 维护的对象。对于每个对象目标，OSB 将创建描述相应对象特征的文件。

只有在很少的情况下才需要直接从文件系统访问管理数据库。OEM、Web 工具和 obtool 接口经常用于访问目录和配置数据。

5.3.4　OSB 用户和类

为了使 OSB 维持多个管理域之间用户身份的一致，OSB 在管理服务器上保存 OSB 用户及其权利的信息。

在管理服务器上，每个 OSB 用户都有一个账户和加密的口令。使用 Web 工具或 obtool，操作系统用户可以输入用户名和口令。而使用加密的 SSL 连接，则由客户端程序传送口令到管理服务器。

当 OSB 安装到管理服务器时，将默认建立管理员用户。在安装期间，也可以建立用于备份和恢复 Oracle 数据库的 Oracle 用户。安装器将为 Oracle 用户分配一个随机口令。通常不需要使用该账户登录 OSB。

5.3.5　操作系统账户

OSB 用户的名称空间与 Linux、Unix 和 Microsoft Windows 用户的名称空间截然不同。因此，如果使用操作系统用户账户 backup_usr 访问管理域中的主机，并且域中具有名为 backup_usr 的 OSB 用户，则这些账户会被分别管理，即使他们的名称是相同的。创建与操作系统用户具有相同名称和口令的 OSB 用户会非常方便。

在创建 OSB 用户时，可以将该用户与 Unix 和 Microsoft Windows 账户关联起来。这种类型的账户一般用作无权限的备份，不需要根权限就可以运行这种备份。有权限的备份和还原操作则使用具有根(Unix)或本地系统(Microsoft Windows)权限的客户端。

如果将要创建名为 backup_usr 的 OSB 用户，并且将其与 Unix 账户 ubackup_usr 和 Microsoft Windows 账户 wbackup_usr 关联，当 backup_usr 使用 backup unprivileged 命令备份客户端时，该作业将在与 backup_usr 关联的操作系统账户下运行。因此，backup_usr 只能备份 Unix 客户端上 ubackup_usr 可访问的文件以及 Microsoft Windows 客户端上 wbackup_usr 可访问的文件。

如果具有“修改管理域配置”的权利，就可以配置 OSB 用户的预授权属性。该权利允许我们事先授权给一些操作系统用户以让他们创建 RMAN 备份或访问 OSB 命令行实用程序。

5.3.6　NDMP 主机

当设置 OSB 用户账户时，可以配置用户以访问 NDMP。可以设置主机以使用用户定义的文本口令、空口令或默认的 NDMP 口令。NDMP 主机的口令与主机(而不是用户)关联。可以配置口令验证方法，例如 MD5 加密或文本。

5.3.7　OSB 权限和类

一个 OSB 类是指授予 OSB 用户的一组已定义的权限。虽然类似于 Unix 组，但 OSB 类具有满足 OSB 特定需求的具体访问权限。如图 5-5 所示，可以将多个用户分配给一个类，而每个用户是一个类的一个成员。

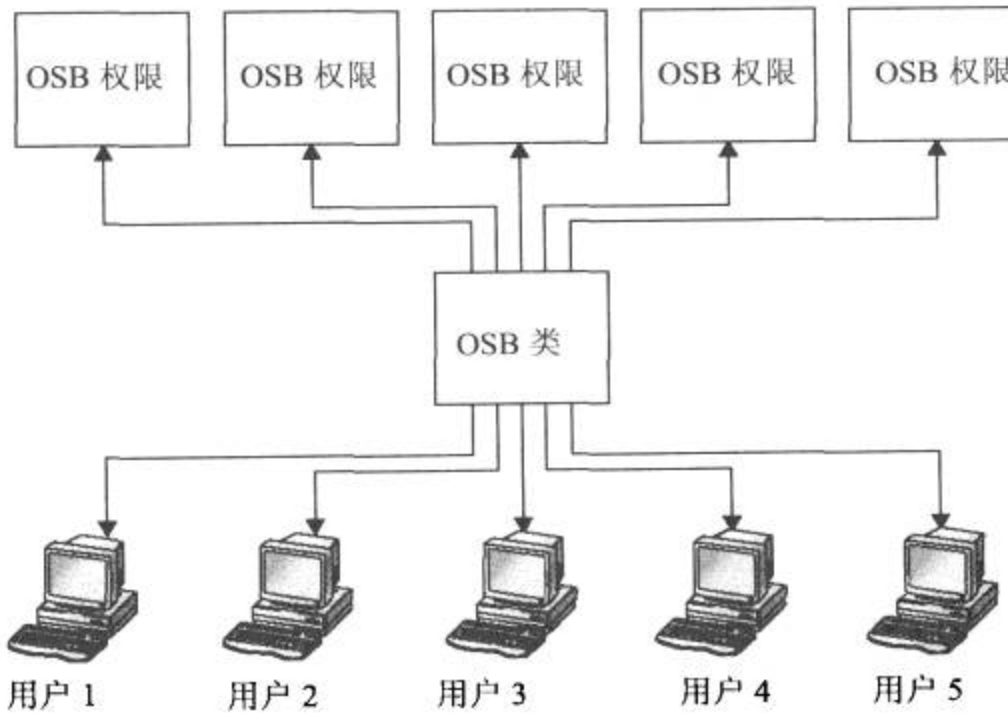


图 5-5　权限和类

下面这些类对于理解 OSB 用户的权限非常重要：

- admin：该类用于域的整体管理，由改变域配置以及完成备份和还原操作所需的所有权利组成。
- monitor：该权限不允许用户接受 e-mail 通知，改变域配置，或执行备份或还原操作。用户仅能访问备份，查看存储设备和域配置信息，显示调度任务列表。

- operator：对于标准的日常操作，该类没有任何配置权利，但包括备份和还原操作所需的所有权利。该权限允许用户控制和查询主存储设备和辅助存储设备的状态。
- Oracle：非常类似于 operator 类，但授予用户改变 Oracle 数据库配置设置和执行 Oracle 数据库备份的权利。该类的成员通常是与 Oracle 数据库软件的操作系统账户对应的 OSB 用户。
- reader：允许成员浏览 OSB 目录。访问者也许只能修改他们自己的账户名称和口令。
- user：分配给用户权利，允许他们以有限的方式与他们的域交互。该类允许用户浏览 OSB 目录中他们自己的数据，并且执行基于用户的还原。

5.4　安装与配置 OSB

因为 OSB 是专门为 RMAN 设计的，将 OSB 与 Oracle 链接在一起非常快速且容易。可以通过 CD-ROM 光盘获得 OSB，或者从 Oracle 技术网络(Oracle Technology Network，OTN)网站上下载 OSB，网址为：

```
http://www.oracle.com/technology/products/secure-backup/index.html
```

下面是安装和配置 OSB 的需求：

- OSB 管理域的每个主机必须运行 TCP/IP。所有主机应具有静态 IP 地址，或者能够保证 DHCP 服务器总是能为主机分配相同的 IP 地址。
- 在 OSB 管理域无重复的主机名，因为索引目录数据基于客户端主机名。
- 对于 Linux 介质服务器，需要该服务器安装有 SCSI Generic(SG)驱动器
- 应用 OSB 的 RAC 集群的每个节点都需要安装 OSB。

RMAN 实习：安装和配置 OSB

实习注意事项

下面的示例使用 Linux 操作系统。从 OTN 中下载 OSB。在 Linux 中，推荐的 OSB 安装目录是/usr/local/oracle/backup。为了简化该示例，管理服务器、介质服务器和客户端安装在同一台计算机上。

步骤 1　以根用户的身份检查系统中是否安装有解压程序。如果没有，则建立一个到 gunzip 实用程序的符号链接：

```
[root@lin32 ~]#  uncompress
-bash: uncompress: command not found
[root@lin32 ~]#  in -s /bin/gunzip /bin/uncompress
```

步骤 2 以根用户的身份创建下载目录，然后对该目录执行 change directory 命令：

```
[root@lin32~]# mkdir download
[root@ lin32~]# cd download/
```

步骤 3 将 OSB 下载到下载目录中，然后解压缩该产品：

```
[root@lin32 download]# ls -l
total 43864
-rw-r--r-- 1 root root 44866571 Jan 19 20:31 osb-10.3.0.1.0_linux32_release.zip
[root@lin32 download]# unzip osb-10.3.0.1.0_linux32_release.zip
Archive: osb-10.3.0.1.0_linux32_release.zip
    creating: osb-10.3.0.1.0_linux32_cdrom090504/
. extracting: osb-10.3.0.1.0_linux32_cdrom090504/OSB.10.3.0.1.0_LINUX32
 .rel
    creating: osb-10.3.0.1.0_linux32_cdrom090504/doc/
    creating: osb-10.3.0.1.0_linux32_cdrom090504/doc/dcommon/
    creating: osb-10.3.0.1.0_linux32_cdrom090504/doc/dcommon/css/
  inflating: osb-10.3.0.1.0_linux32_cdrom090504/doc/dcommon/css/blafdoc.css
  inflating: osb-10.3.0.1.0_linux32_cdrom090504/doc/dcommon/css/bp_layout.css
...
  inflating: osb-10.3.0.1.0_linux32_cdrom090504/welcome.html
  inflating: osb-10.3.0.1.0_linux32_cdrom090504/doc.tar
```

步骤 4 创建安装程序存放 OSB 文件的目录：

```
[root@lin32 download ]# mkdir -p /usr/local/oracle/backup
```

步骤 5 对该 OSB 目标执行更改 directory 命令并运行安装命令：

```
[root@lin32 download]# cd/usr/local/oracle/backup/
[root@lin32 backup]#
       /root/download/osb-10.3.0.1.0_linux32_cdrom090504/setup
```

返回如下输出：

```
Welcome to Oracle's setup program for Oracle Secure Backup. This program loads
Oracle Secure Backup software from the CD-ROM to a filesystem directory of your
choosing.
This CD-ROM contains Oracle Secure Backup version 10.3.0.1.0_LINUX32.
Please wait a moment while I learn about this host... done.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   1. linux32
      administrative server, media server, client
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Loading Oracle Secure Backup installation tools... done.
Loading linux32 administrative server, media server, client... done.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Oracle Secure Backup has installed a new obparameters file.
Your previous version has been saved as install/obparameters.savedbysetup.
```

```
Any changes you have made to the previous version must be made to the new
obparameters file.
Would you like the opportunity to edit the obparameters file
Please answer 'yes' or 'no' [no]:
```

步骤 6　选择默认参数，单击 ENTER 键选择默认选项。返回如下的输出：

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - -
Loading of Oracle Secure Backup software from CD-ROM is complete.
You may unmount and remove the CD-ROM.
Would you like to continue Oracle Secure Backup installation with 'installob'
now?
(The Oracle Secure Backup Installation Guide contains complete information about
installob.)
Please answer 'yes' or 'no' [yes]:
```

步骤 7　再一次单击 ENTER 键选择默认选项。然后可以看到如下输出：

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - -
Welcome to installob, Oracle Secure Backup's installation program.
For most questions, a default answer appears enclosed in square brackets.
Press Enter to select this answer.
Please wait a few seconds while I learn about this machine... done.
Have you already reviewed and customized install/obparameters for your Oracle
Secure Backup installation [yes]?
```

步骤 8　再一次单击 ENTER 键选择默认选项选择默认参数。然后可以看到如下输出：

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Oracle Secure Backup is not yet installed on this machine.
Oracle Secure Backup's Web server has been loaded, but is not yet configured.
Choose from one of the following options. The option you choose defines the
software components to be installed.
Configuration of this host is required after installation completes.
You can install the software on this host in one of the following ways:
    (a) administrative server, media server and client
    (b) media server and client
    (c) client
If you are not sure which option to choose, please refer to the Oracle Secure
Backup Installation Guide. (a,b or c) [a]?
```

步骤 9　将在同一个服务器上安装 OSB 的三个组件，再一次按 ENTER 键选择默认选项。然后可以看到如下输出：

```
Beginning the installation. This will take just a minute and will produce
several lines of informational output.
Installing Oracle Secure Backup on lin32 (Linux version 2.6.18-53.el5)
You must now enter a password for the Oracle Secure Backup encryption key store.
```

```
Oracle suggests you choose a password of at least 8 characters in length,

containing a mixture of alphabetic and numeric characters.
Please enter the key store password:
Re-type password for verification:
```

步骤 10 两次输入 OSB 加密口令。不会显示口令。然后可以看到如下输出:

```
You must now enter a password for the Oracle Secure Backup 'admin' user. Oracle
suggests you choose a password of at least 8 characters in length, containing a
mixture of alphabetic and numeric characters.
Please enter the admin password:
Re-type password for verification:
```

步骤 11 输入管理员口令两次。不会显示口令。然后可以看到如下输出:

```
You should now enter an email address for the Oracle Secure Backup 'admin' user.
Oracle Secure Backup uses this email address to send job summary reports and
to notify the user when a job requires input. If you leave this blank, you
can set it later using the obtool's 'chuser' command.
Please enter the admin email address:
```

步骤 12 不用填写 E-mail 地址。然后可以看到如下输出:

```
generating links for admin installation with Web server
updating /etc/ld.so.conf
checking Oracle Secure Backup's configuration file (/etc/obconfig)
setting Oracle Secure Backup directory to /usr/local/oracle/backup in /etc/
obconfig
setting local database directory to /usr/etc/ob in /etc/obconfig
setting temp directory to /usr/tmp in /etc/obconfig
setting administrative directory to /usr/local/oracle/backup/admin in /etc/
obconfig
protecting the Oracle Secure Backup directory
creating /etc/rc.d/init.d/observiced
activating observiced via chkconfig
initializing the administrative domain
******************************** N O T E ********************************
On Linux systems Oracle recommends that you answer no to the next two questions.
The preferred mode of operation on Linux systems is to use the /dev/sg devices for
attach points as described in the 'ReadMe' and in the 'Installation and
Configuration Guide'.
Is lin32 connected to any tape libraries that you'd like to use with Oracle
Secure Backup [no]?
Is lin32 connected to any tape drives that you'd like to use with Oracle Secure
Backup [no]?
```

步骤 13 由于在该例中使用的是 Linux 系统，按照 Oracle 的建议应选择“no”，选

择以后配置介质服务器。将返回以下结果:

```
Installation summary:
    Installation Host          OS      Driver       OS Move      Reboot

          Mode     Name          Name     Installed?    Required?    Required?
     admin        lin32          Linux    no            no           no
Oracle Secure Backup is now ready for your use.
```

通过以上步骤，OSB 管理服务器、介质服务器和客户端均已安装。OSB Web 工具用于配置磁带库和磁带驱动器。下面将建立 Oracle 用户和数据库备份存储选择器以确保对 Oracle 数据库的备份。

步骤 14　以管理员用户身份通过使用 https://<administrative server>链接并登录到 OSB Web 工具上。进入 Configure 页面，单击 Users 链接，单击 Add 按钮并增加 oracle 用户，如图 5-6 所示。

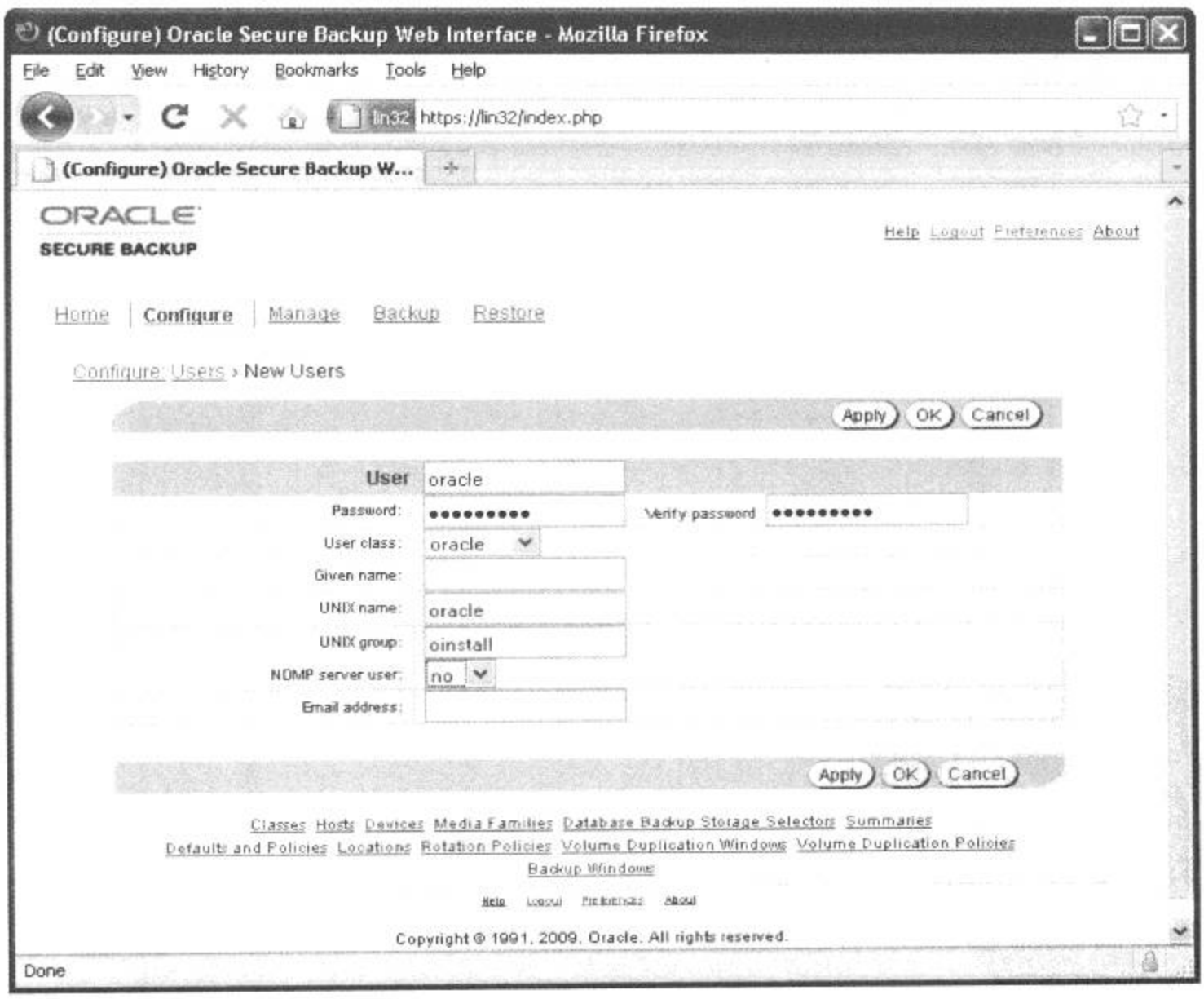


图 5-6　New Users 页面

步骤 15　增加 Oracle 用户后，单击 Edit 按钮，更改 Preauthorized Access，如图 5-7

所示。

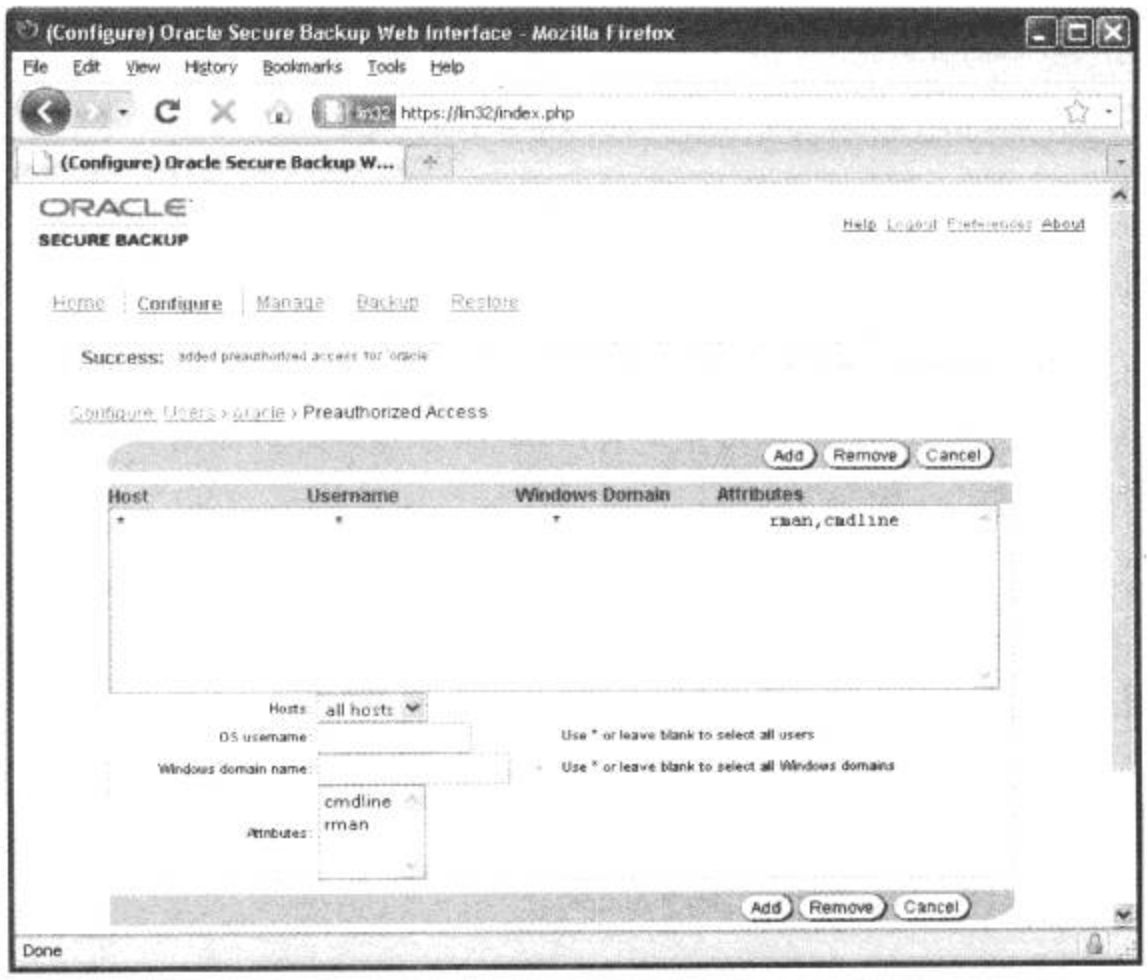


图 5-7　Preauthorized Access 页面

步骤 16　完成以上操作后，将会看到 admin 用户和 Oracle 用户，如图 5-8 所示。

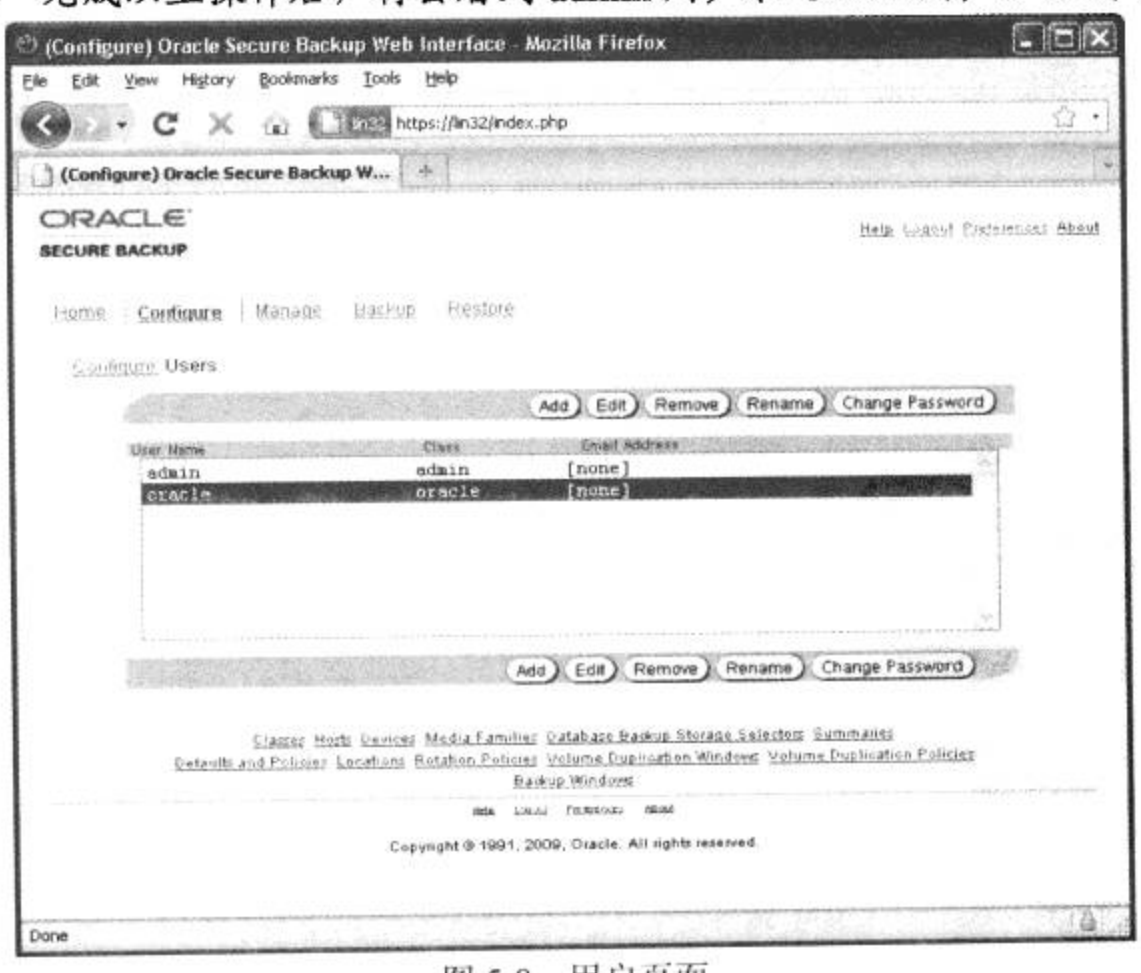


图 5-8　用户页面

步骤 17　返回 Configure 页面：配置主机。确保服务器拥有 mediaserver 角色，如图 5-9 所示。

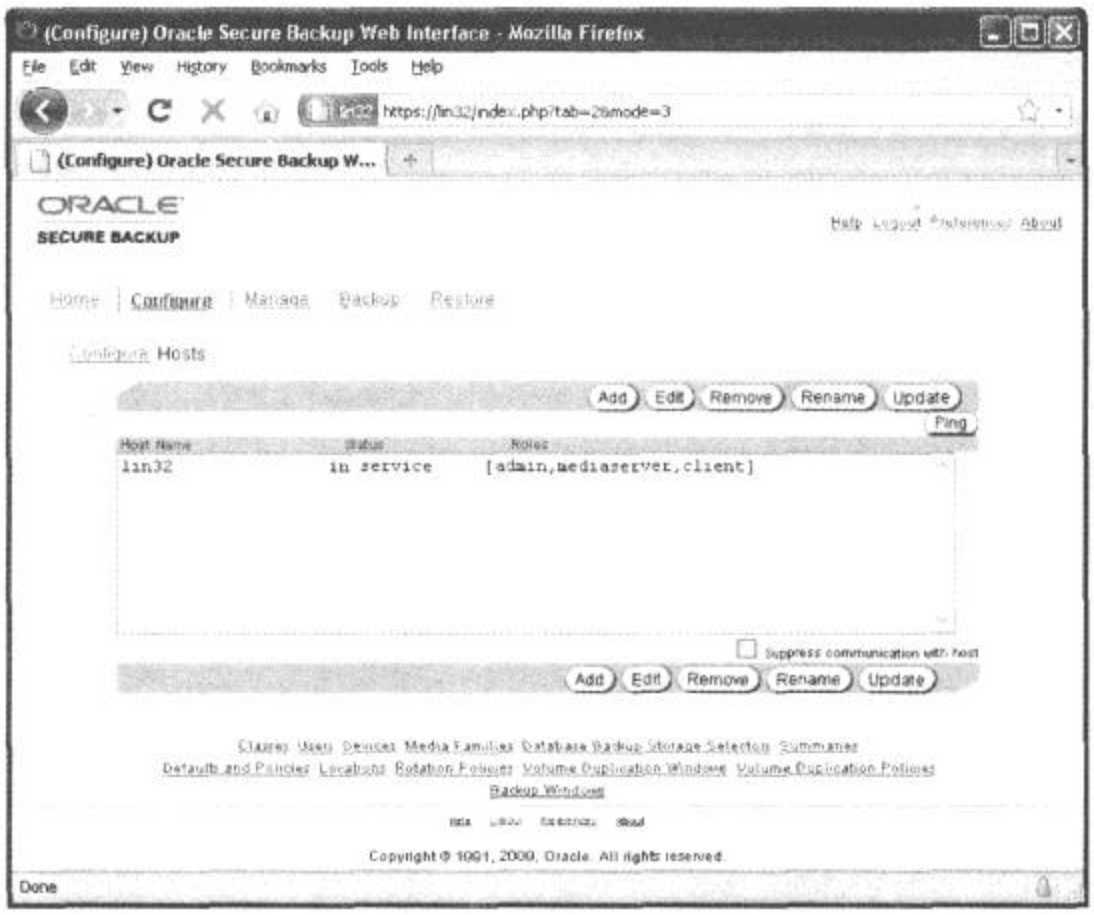


图 5-9　Configure 页面

步骤 18　增加存储选择器，单击 Configure 页面底端的 Database Backup Storage Selector 链接，单击 Add 按钮，按照图 5-10 所示填写相关字段。

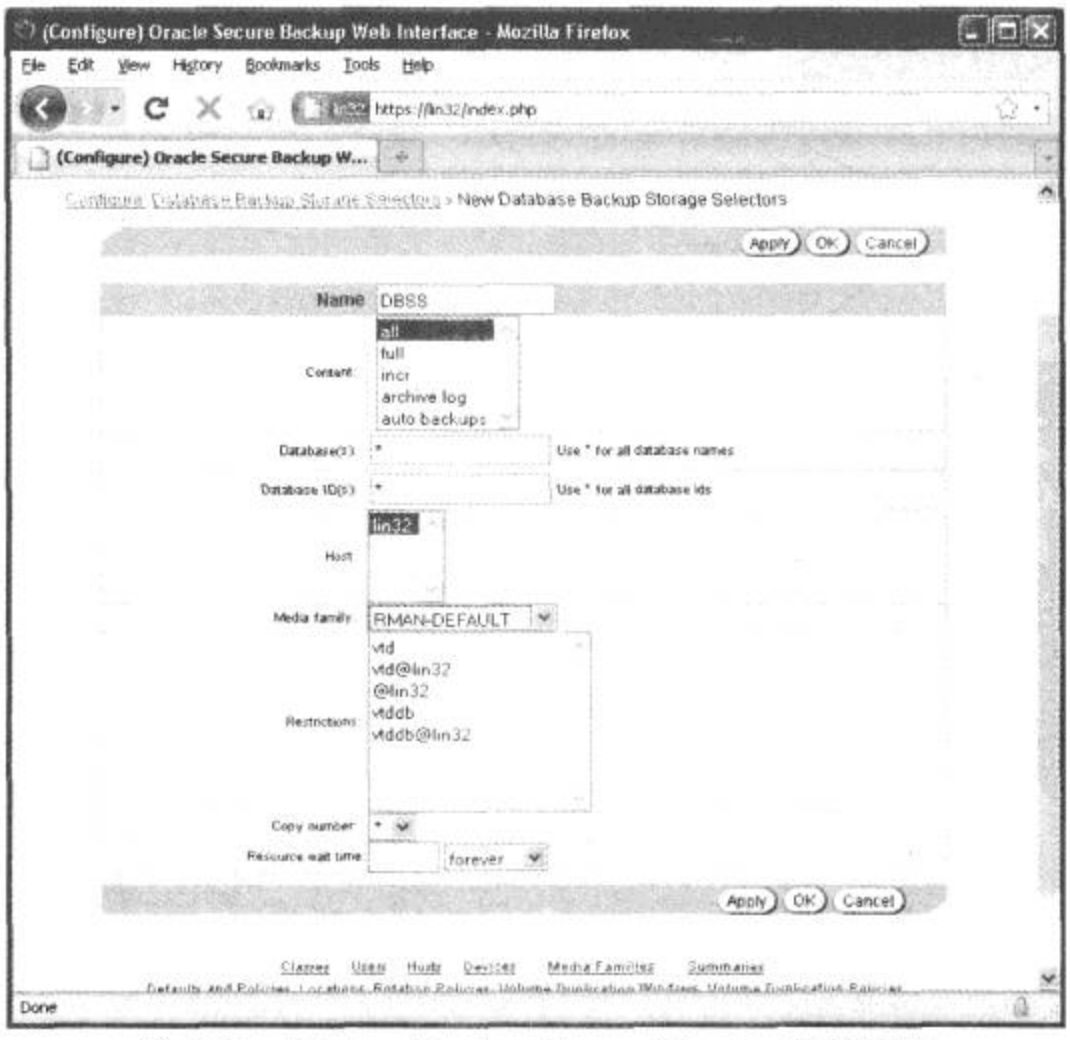


图 5-10　Database Backup Storage Selector 链接页面

步骤 19　单击 OK 按钮，存储将会如图 5-11 所示建立。

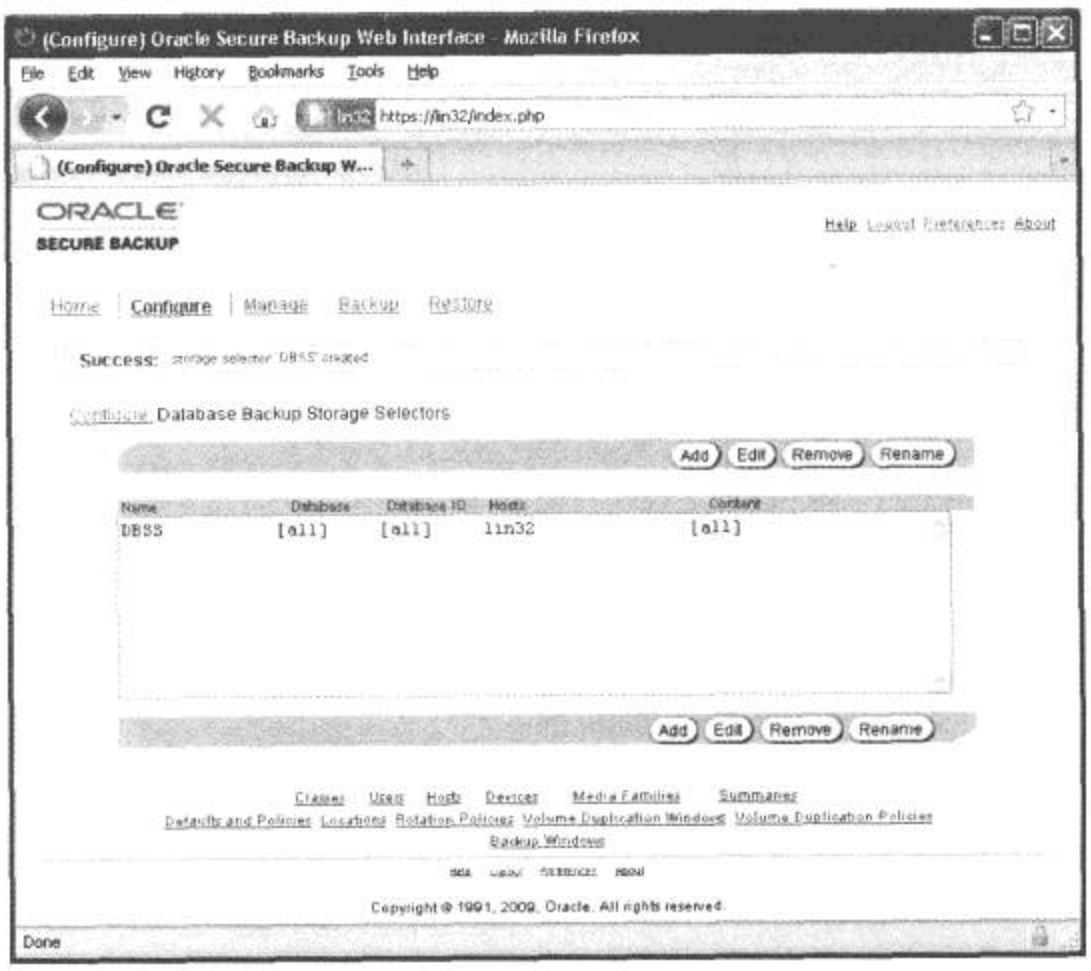


图 5-11　Database Backup Storage Selectors 页面

步骤 20　开始为使用 OSB 配置 OEM。与数据库建立连接并进入 OEM 的 Availability 选项卡。单击 Backup Settings 按钮，进入图 5-12 所示的页面底端。

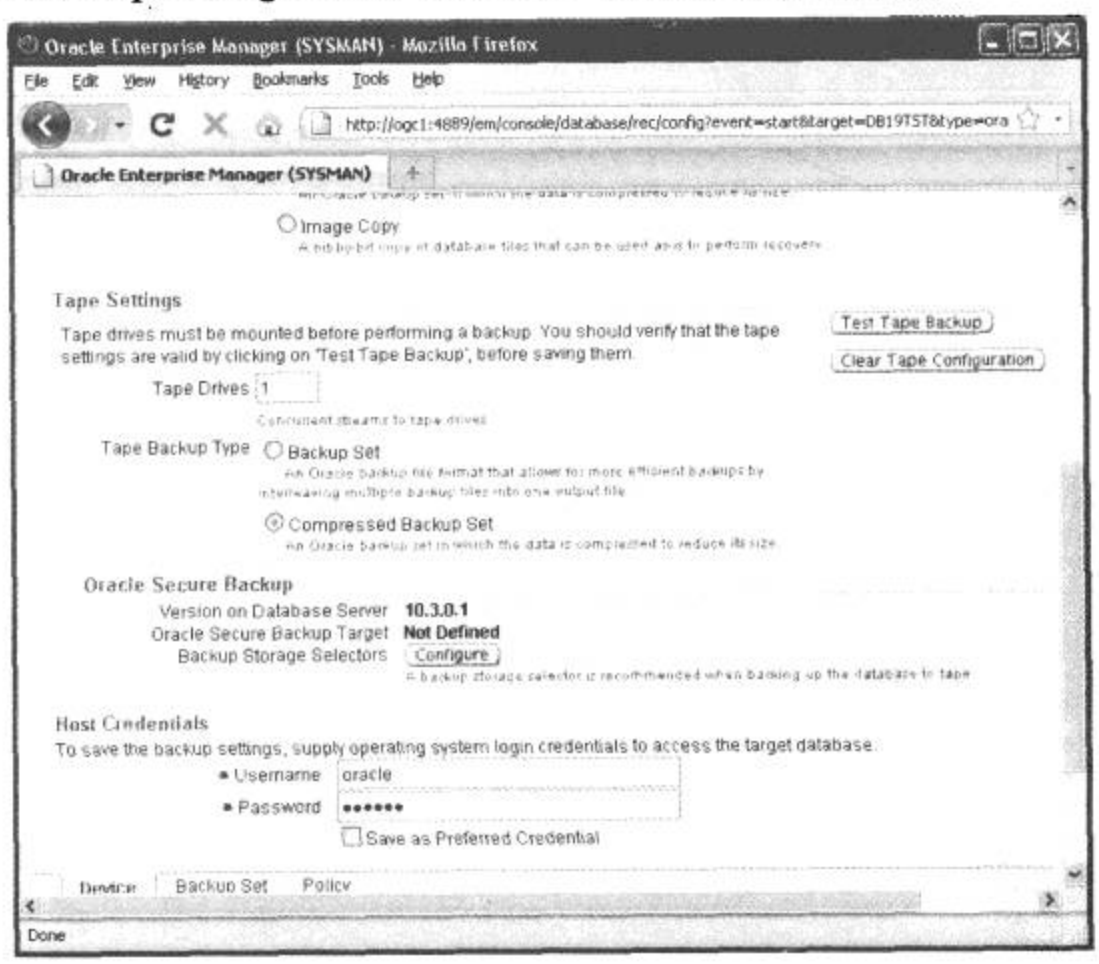


图 5-12　Backup Settings 页面底端

步骤 21　单击 Configure 以配置 OSB 目标，在 Specify Oracle Secure Backup Target 页面，单击 Add 按钮然后进入主机，如图 5-13 所示。

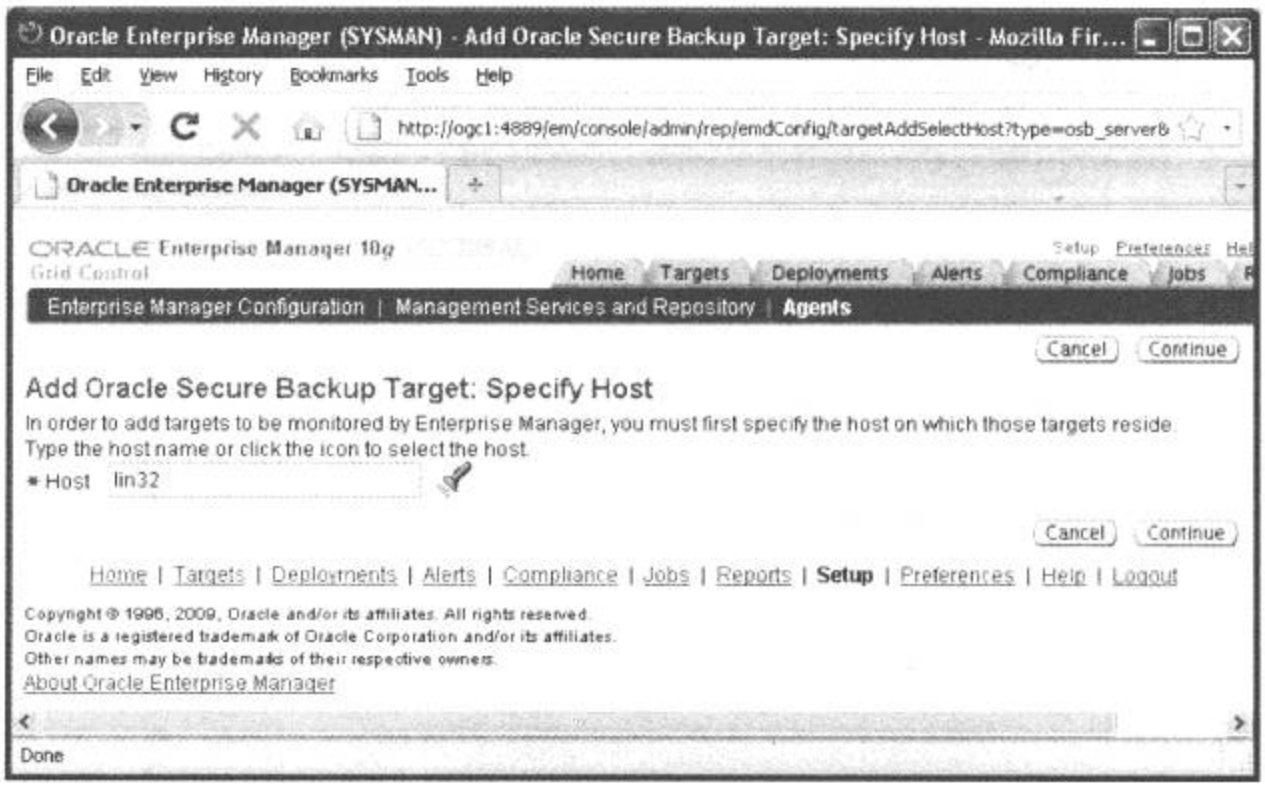


图 5-13　Specify Oracle Secure Backup Target 页面

步骤 22　单击 Continue 按钮，输入如图 5-14 所示的值。

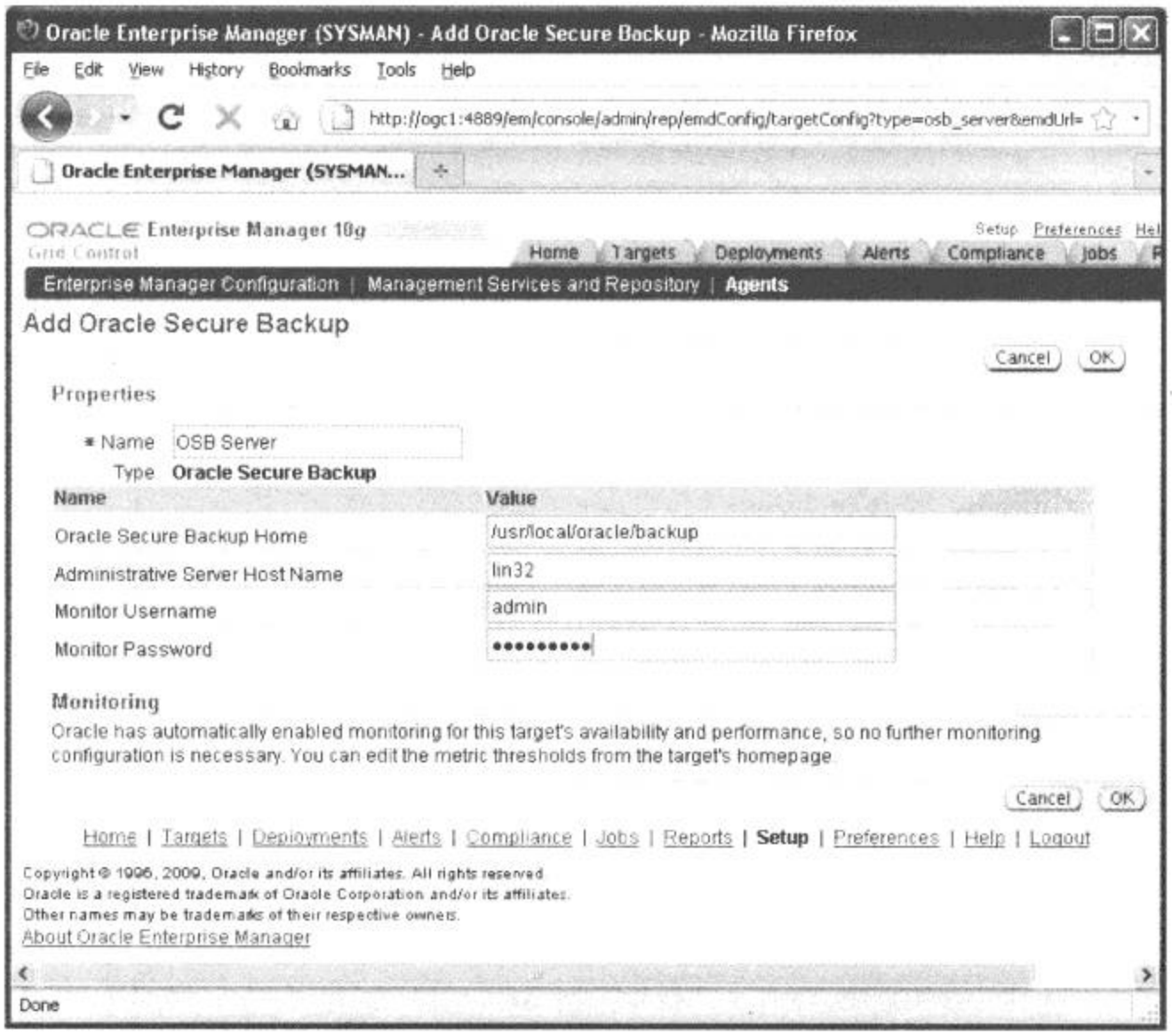


图 5-14　Add Oracle Secure Backup 页面

步骤 23　单击两次 OK 按钮后，将进入 Backup Storage Selectors 页面，如图 5-15 所示。

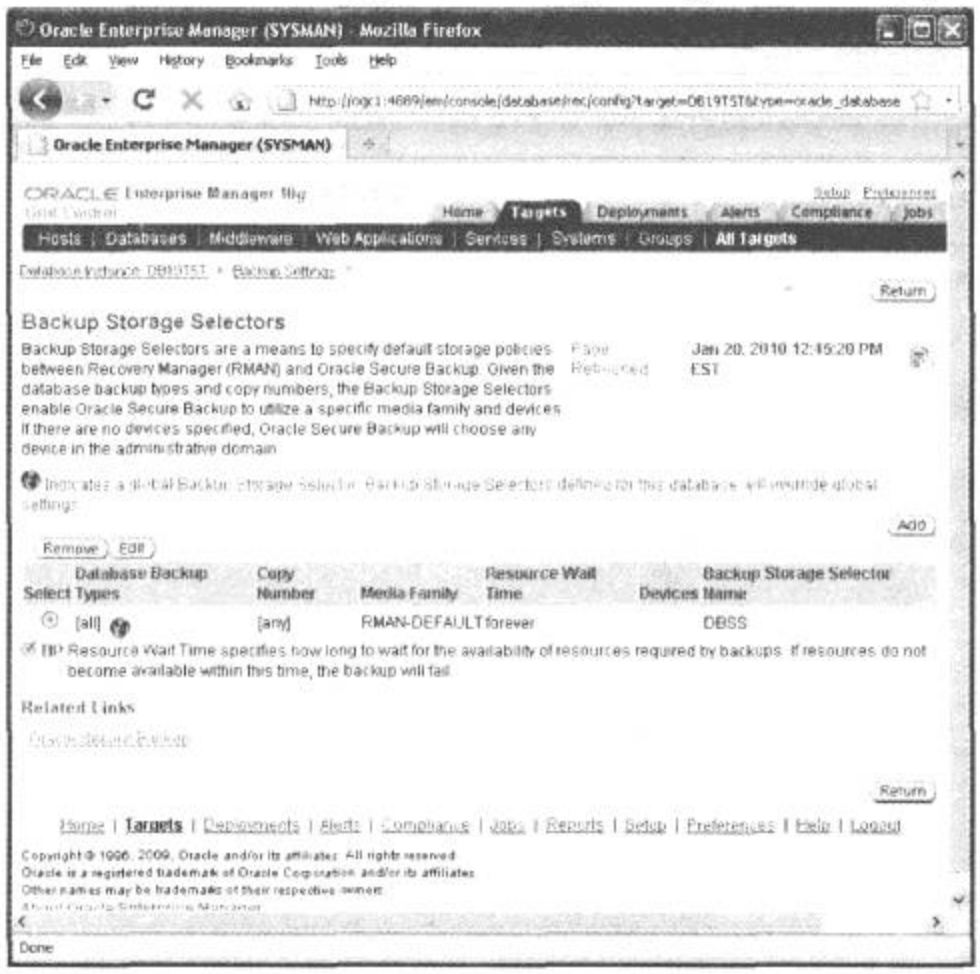


图 5-15　Backup Storage Selectors 页面

步骤 24　单击 Return 按钮，OSB 服务器目标设置为将 Oracle 数据库备份到磁带上。如图 5-16 所示。OSB 管理服务器被配置为 OEM 目标，因此可以使用 OEM 管理。

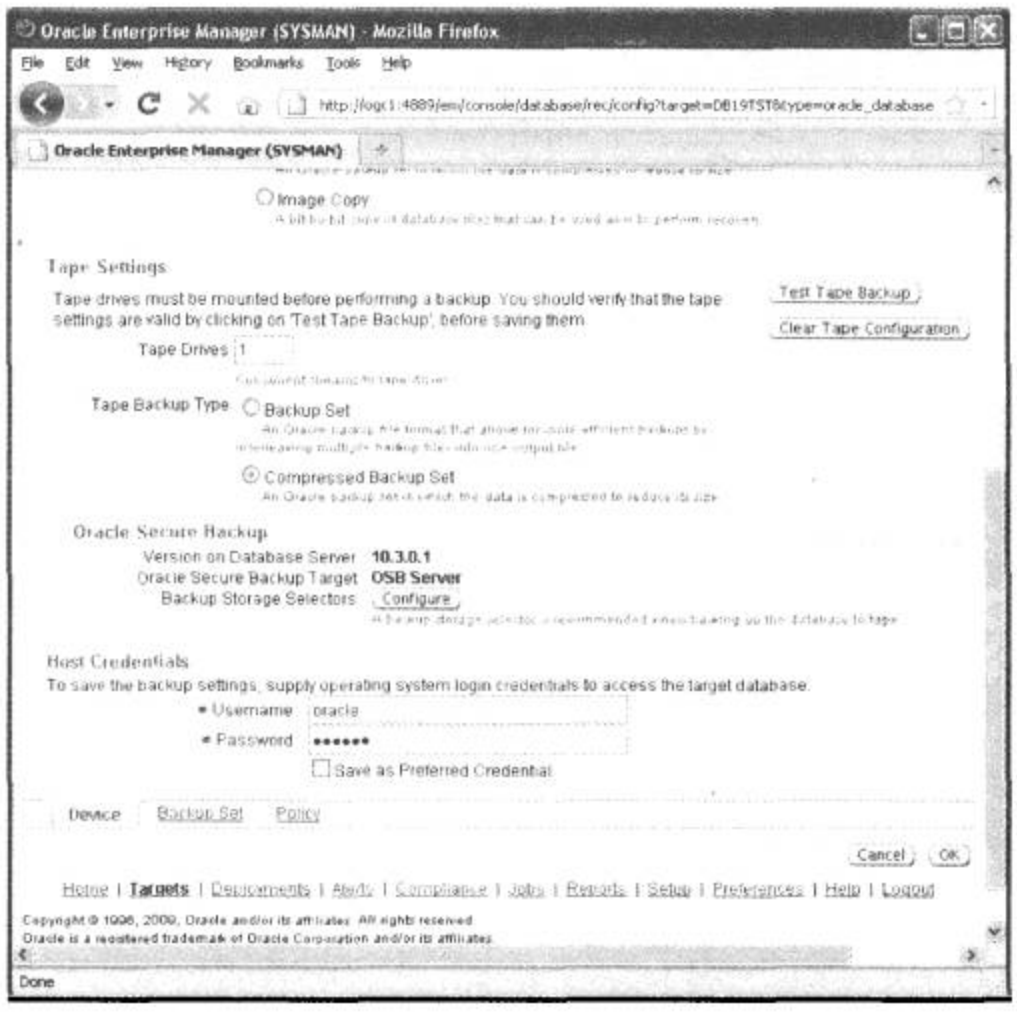


图 5-16　Return 页面

步骤 25　在 OEM 的 All Targets 页面上寻找 OSB 服务器。如图 5-17 所示。

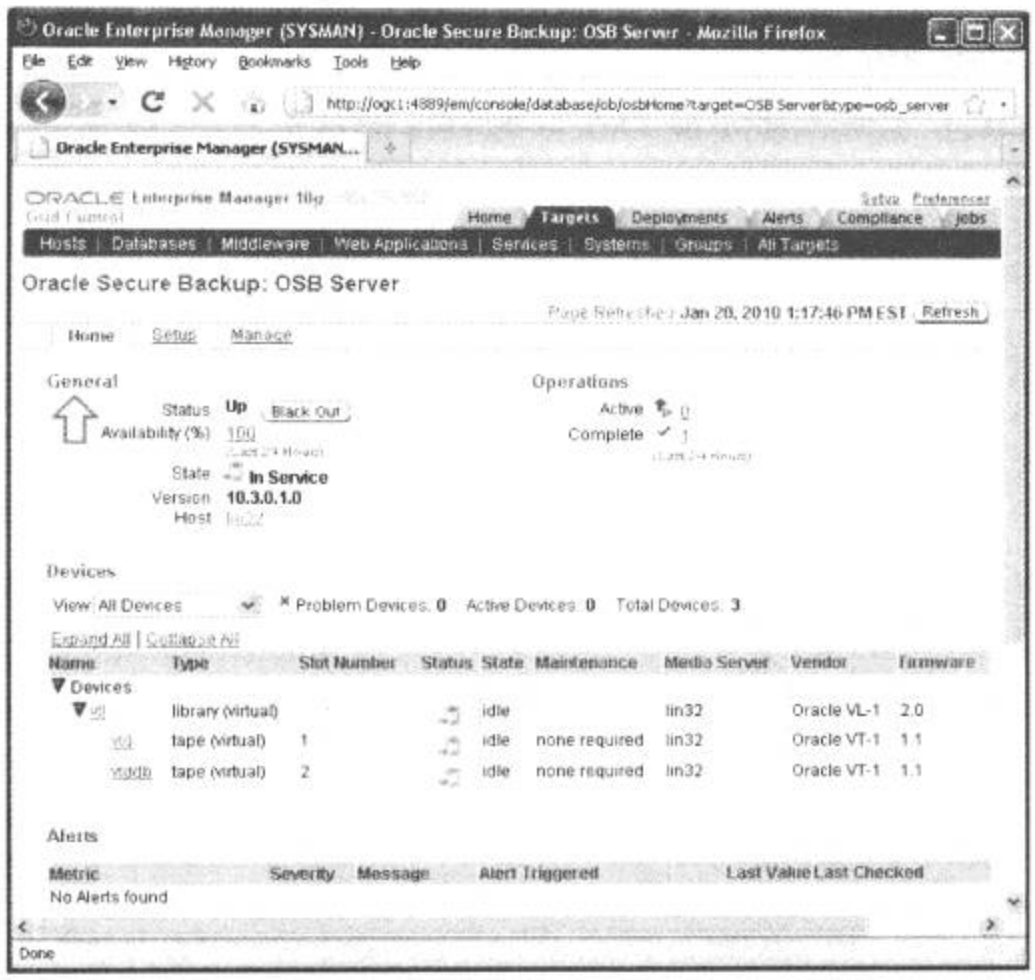


图 5-17　All Targets 页面

步骤 26　单击 Setup 选项卡配置 OSB 服务器。OSB 设备可以在 Devices 页面上配置。如图 5-18 所示。

单击 Manager 选项卡，可以在此调度文件系统数据的备份和还原工作。

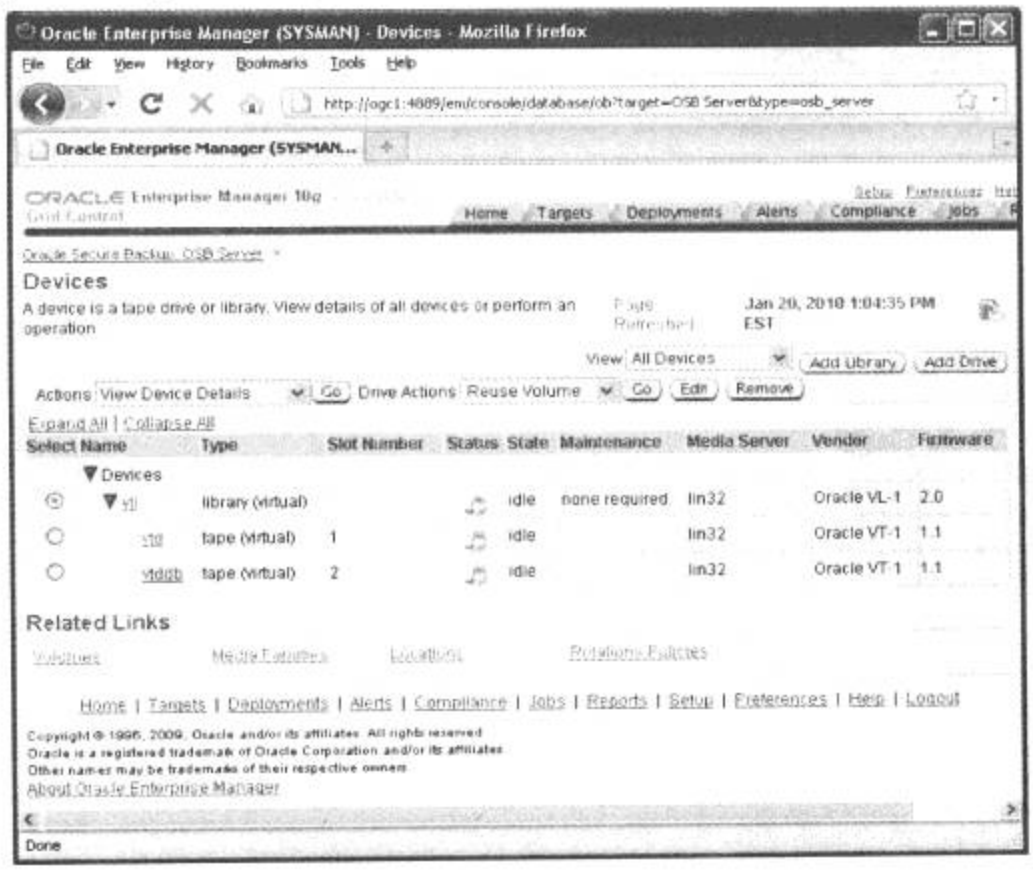


图 5-18　Devices 页面

5.5　使用 OSB 备份 Oracle 数据库及文件系统数据

使用 OSB Web 工具不能实现 Oracle 数据库的备份与还原操作。因此，建议使用 OEM 作为集中式接口，用以调度 Oracle 数据库和文件系统数据的备份和恢复操作。

RMAN 实习：Oracle 数据库及文件系统数据的备份调度

实习注意事项

该实习调度 OSB Oracle 数据库及文件系统数据备份。首先，完成一次完整的 Oracle 数据库备份操作。

步骤 1　连接到数据库，进入 OEM 的 Availability 页面。单击 Schedule Backup 按钮。然后，选择 Whole Database 选项并单击 Schedule Customized Backup 按钮。在该页面上，可以选择不同的备份操作。单击 Next 按钮，在 Settings 页面上，选择 Tape 选项，单击 Next 按钮，如图 5-19 所示。

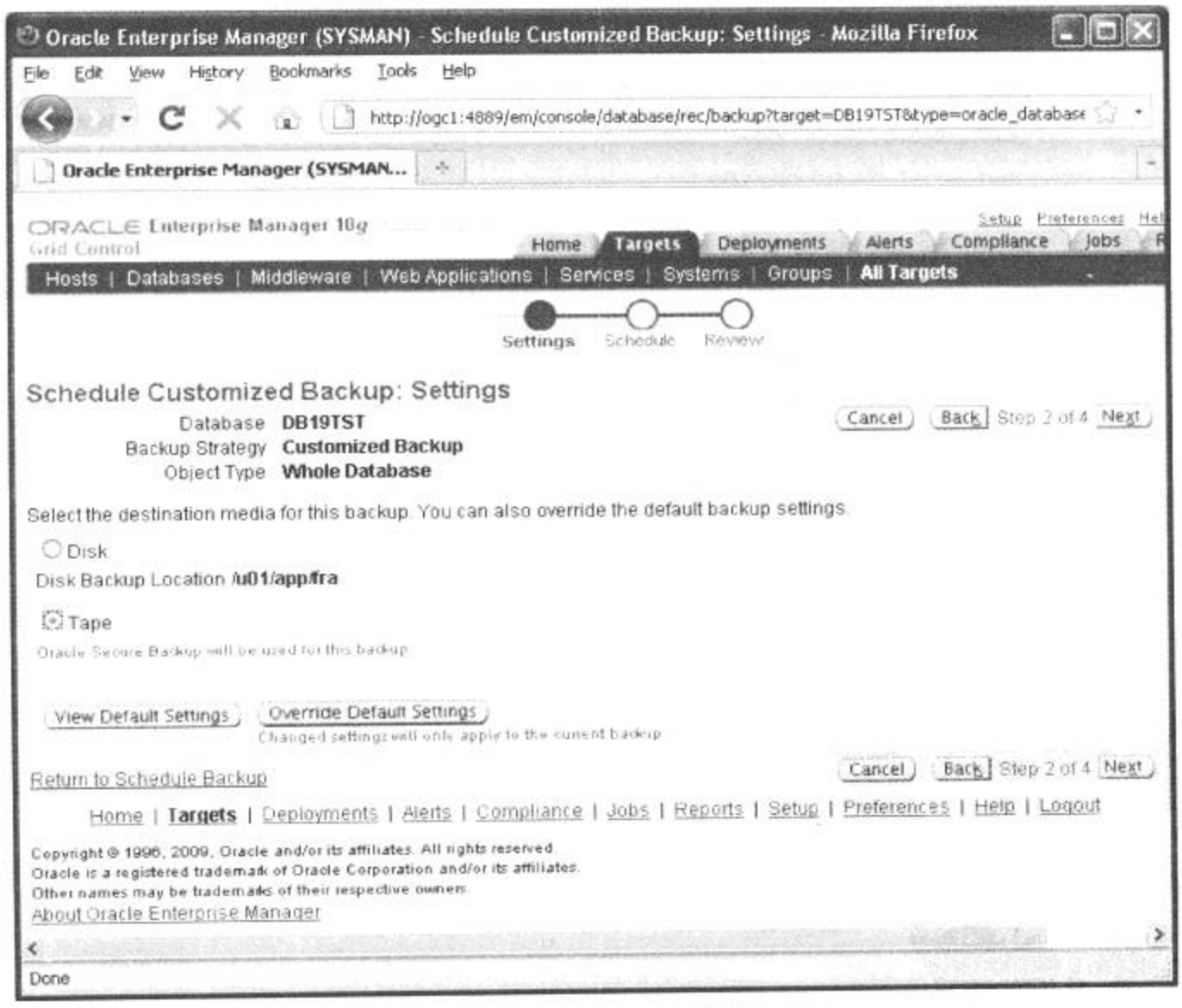


图 5-19　Settings 页面

步骤 2　将操作类型选择为 one-time 类型，浏览被调度的任务。然后单击 Submit Job 按钮提交任务，如图 5-20 所示。

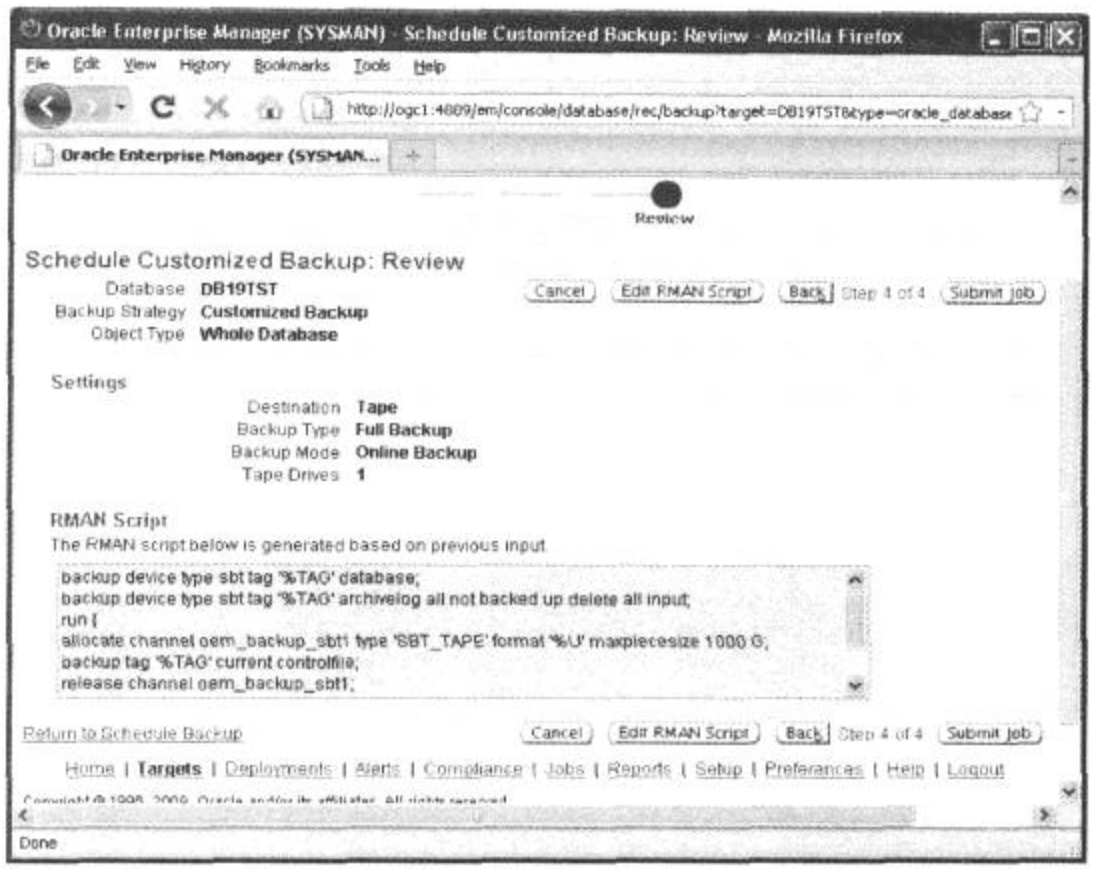


图 5-20　Review 页面

步骤 3　OEM 将会标记已提交的任务。可以单击 View Job 按钮查看任务的状态，如图 5-21 所示。

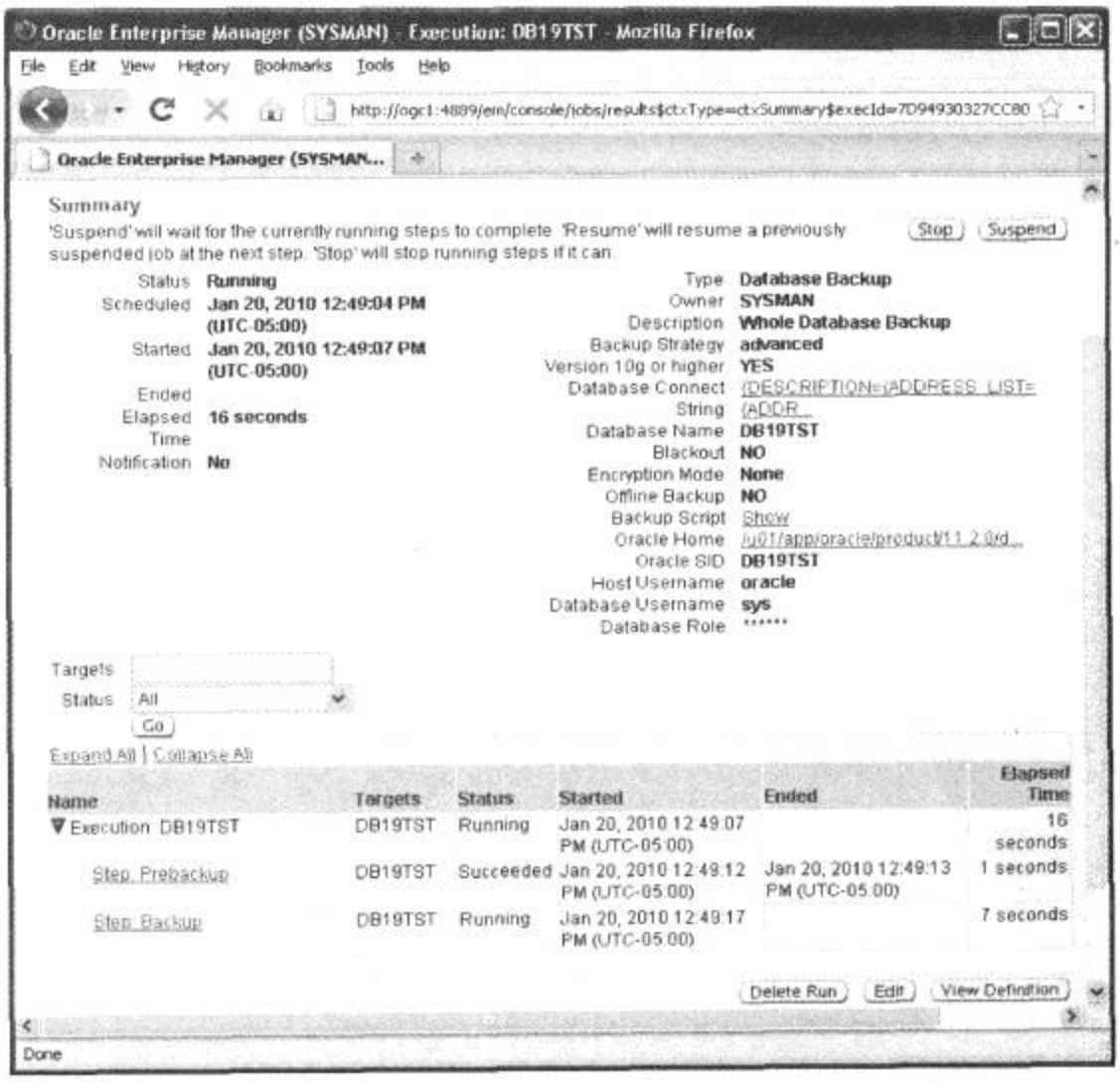


图 5-21　View Job 页面

步骤 4　正在执行的操作可以在 Jobs 页面中浏览，如图 5-22 所示。

图 5-22　Jobs 页面

现在，可以开始备份 listener.ora 文件。

步骤 5　在 OEM OSB 服务器的管理页面中，单击 Schedule Backup 选项，然后选择 Specify Hosts，Directories 和 Files，如图 5-23 所示。

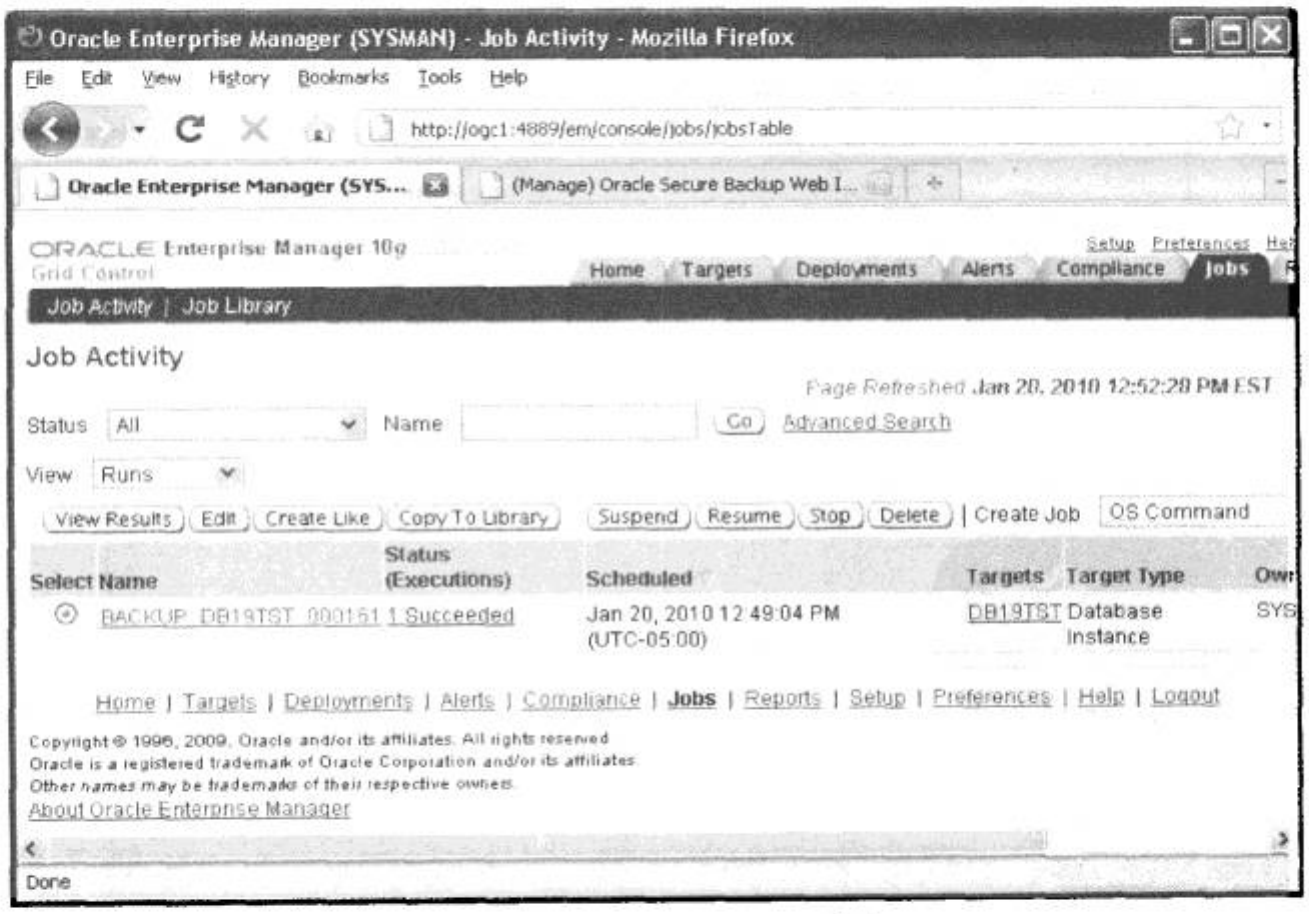


图 5-23　Schedule Backup 页面

步骤 6　将 listener.ora 文件增加到备份中。如图 5-24 所示。

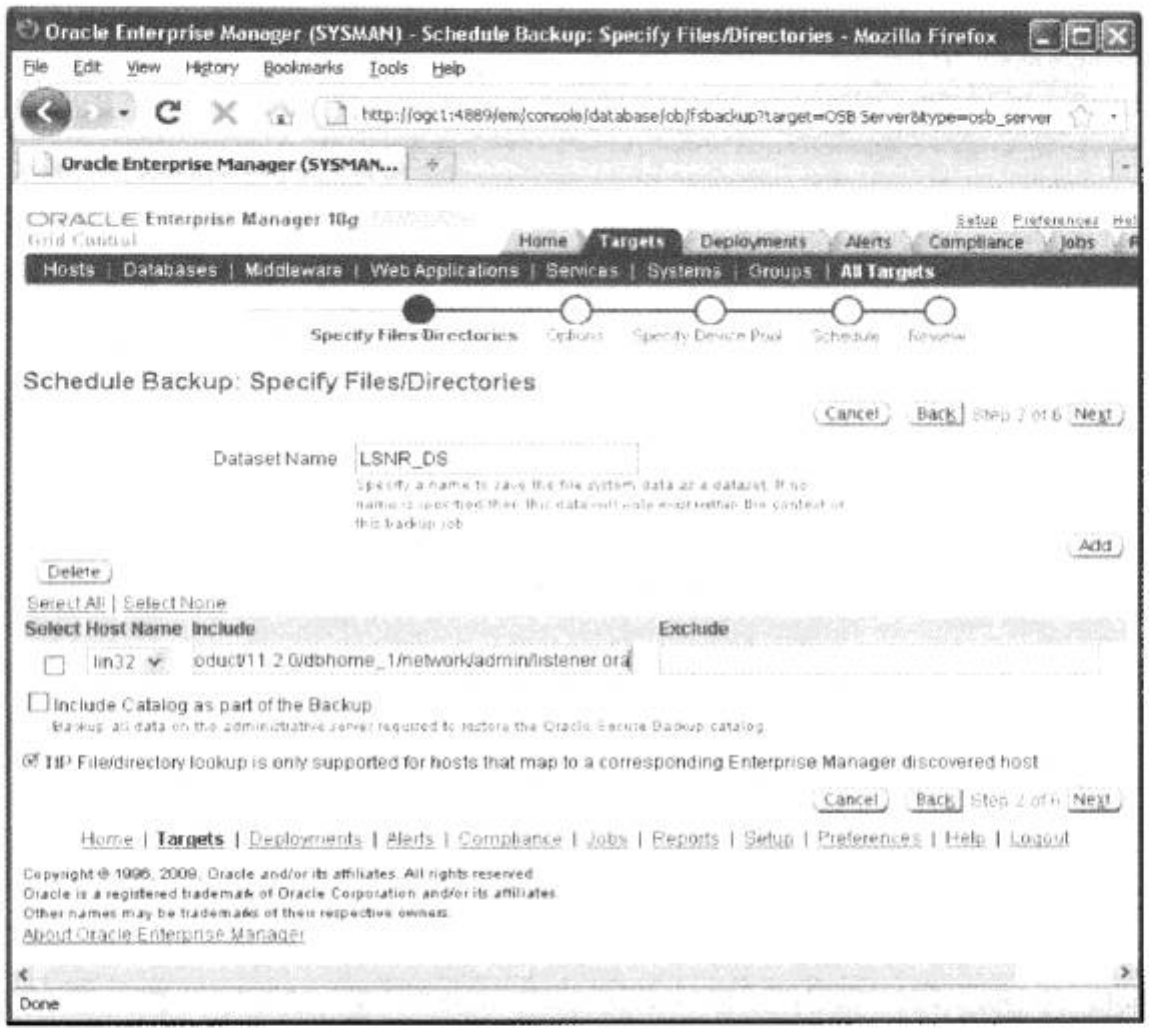


图 5-24　Specify Files/Directories 页面

步骤 7　单击 Next 按钮，检查授权用户使根用户运行了该操作。再次单击 Next 按钮。在 Specify Device Pool 页面中，可以为备份定义磁带驱动器，如图 5-25 所示。

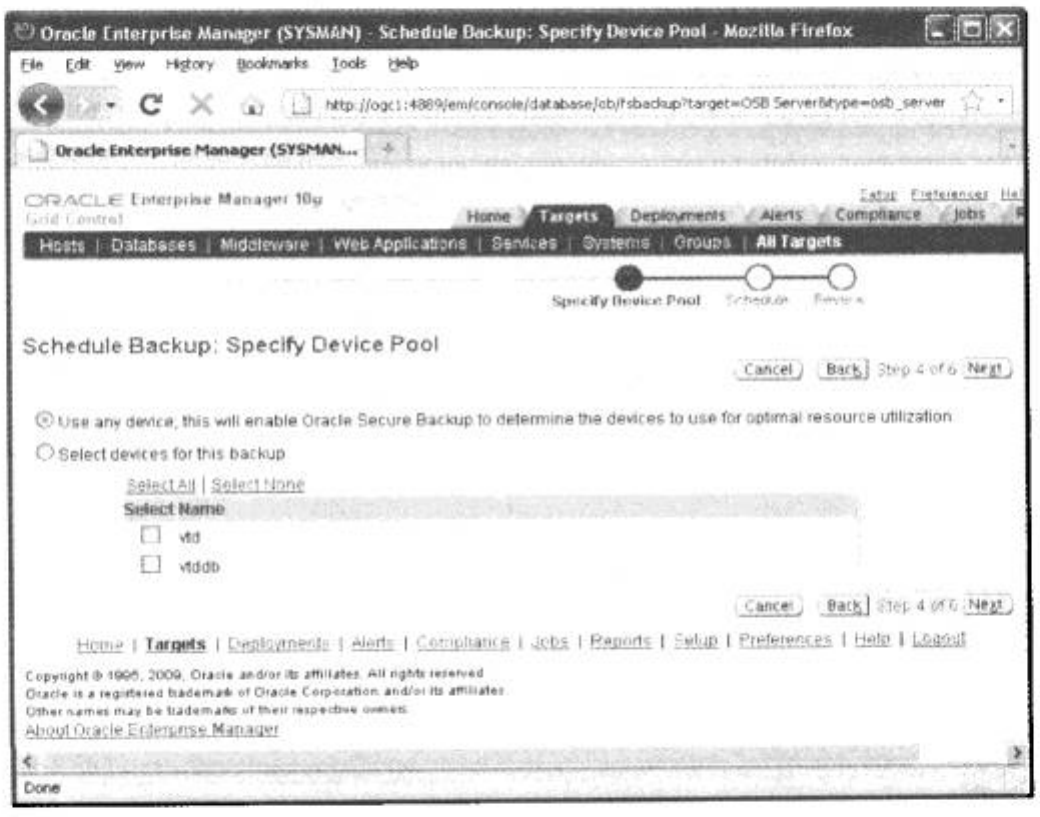


图 5-25　Specify Device Pool 页面

步骤 8　单击 Next 按钮，选择 Job Schedule 选项，然后再次单击 Next 按钮，浏览备份操作，如图 5-26 所示。

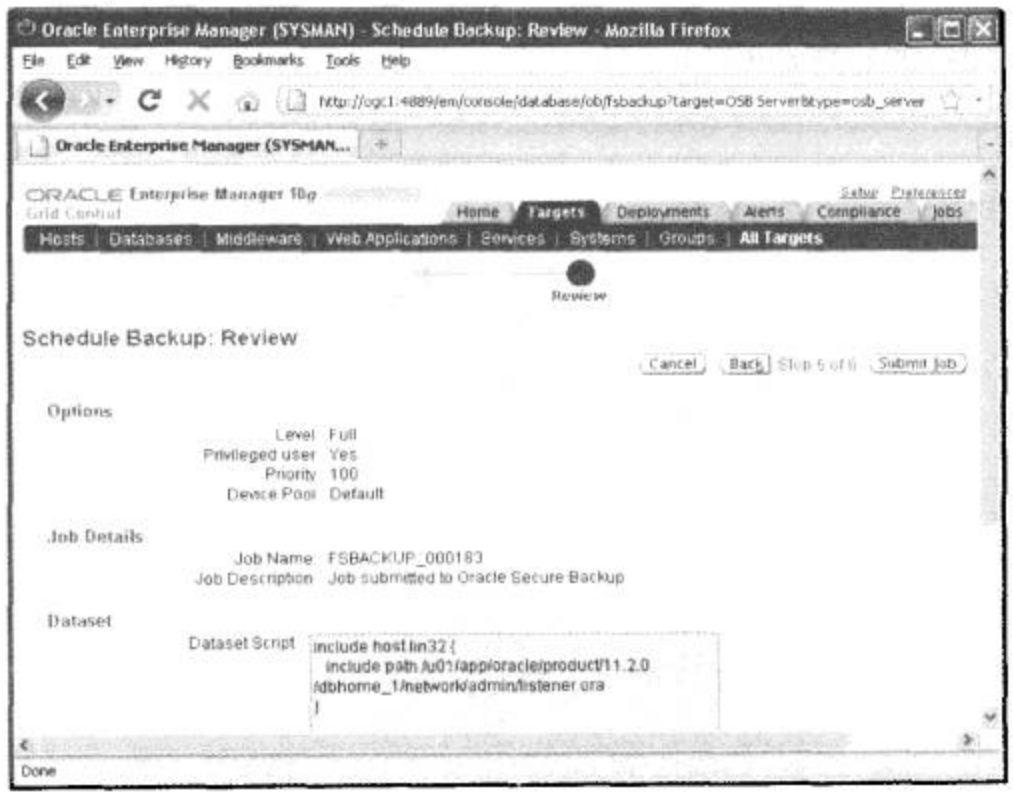


图 5-26　Review 页面

步骤 9　提交操作后，可以看见 OEM 确认信息。在 Jobs 页面可以看到操作的执行状态，如图 5-27 所示。

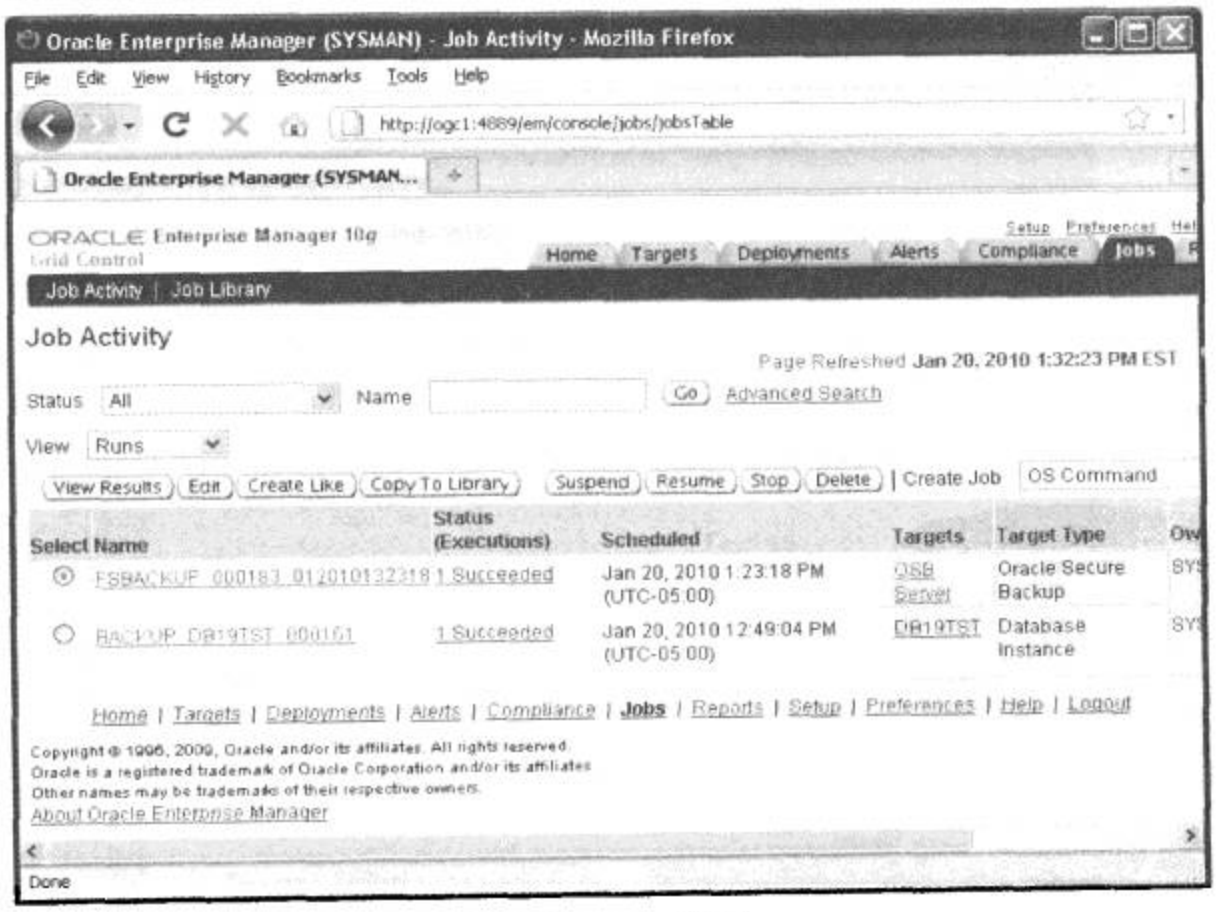


图 5-27　Jobs 页面

5.6　使用 OSB 云模块备份数据库

OSB 提供使用存储云的功能。例如利用 Amazon S3，作为异地备份存储目标。在使用 Amazon S3 作为备份存储时，需要注册到 Amazon S3 服务并且获得 Access Key ID 以及 Secret Access Key。获得访问标识后，可以在 OSB 云模块安装期间配置 OSB。可以从 OTN 下载 OSB 云模块安装工具，地址为：http://www.oracle.com/technology/software/ tech/cloud/index.html.

RMAN 实习：安装 OSB 云模块并使用其执行 OSB 备份

实习注意

该实习安装 OSB 云模块并调度执行 OSB Oracle 的数据库备份。

步骤 1　下载并安装 OSB 云模块：

```
[oracle@lin32 distrib]$ ls -l
total 2428
-rw-r--r-- 1 oracle oinstall 2480195 Jan 19 18:13 osbws_installer.zip
[oracle@lin32 distrib]$ unzip osbws_installer.zip
Archive: osbws_installer.zip
  inflating: osbws_readme.txt
  inflating: osbws_install.jar
[oracle@lin32 distrib]$ vi osbws_install.sh
[oracle@lin32 distrib]$ cat osbws_install.sh
java -jar osbws_install.jar -AWSID <Access Key ID> -AWSKey <Secret Access Key> -
otnUser alisher.yuldashev@<mail_server>.com -otnPass <OTN password> -
walletDir
$ORACLE_HOME/dbs/osbws_wallet
[oracle@lin32 distrib]$ chmod u+x osbws_install.sh
[oracle@lin32 distrib]$ ./osbws_install.sh
Oracle Secure Backup Database Web-Service Install Tool
OTN userid is valid.
AWS credentials are valid.
Creating new registration for this S3 user.
Created new log bucket.
Registration ID: <Registration ID>
S3 Logging Bucket: oracle-log-alisher-1
Validating log bucket location ...
Validating license file ...
Create credential oracle.security.client.connect_string1
OSB web-services wallet created in directory
/u01/app/oracle/product/11.2.0/dbhome_1/dbs/osbws_wallet.
OSB web-services initialization file
/u01/app/oracle/product/11.2.0/dbhome_1/
dbs/osbwsDB19TST
.ora created.
```

步骤 2　连接到数据库，进入 OEM 的 Availability 页面，单击 Schedule Backup 按钮。

然后，选择 Whole Database 选项并单击 Schedule Customized Backup 按钮。建议异地存储时使用加密备份方法，如图 5-28 所示。

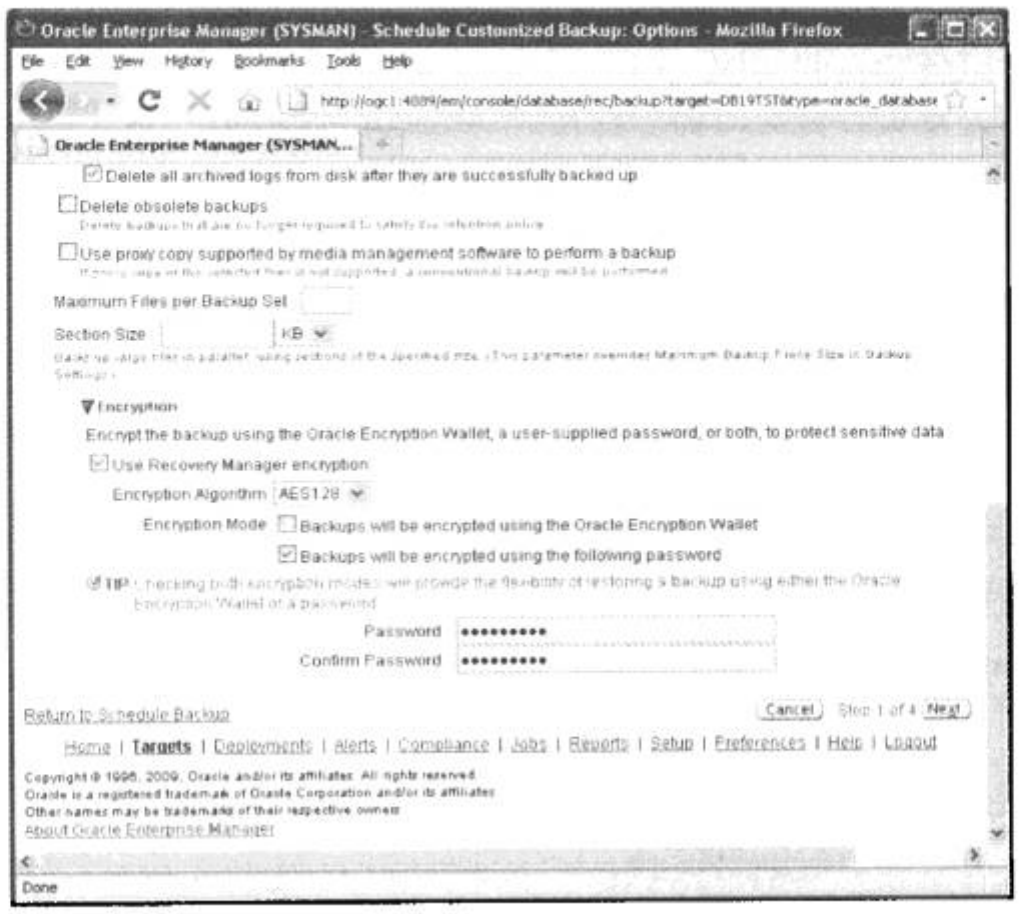


图 5-28　Schedule Customized Backup 页面

步骤 3　在 Settings 页面上，选择 Tape 选项并单击 Next 按钮，如图 5-29 所示。

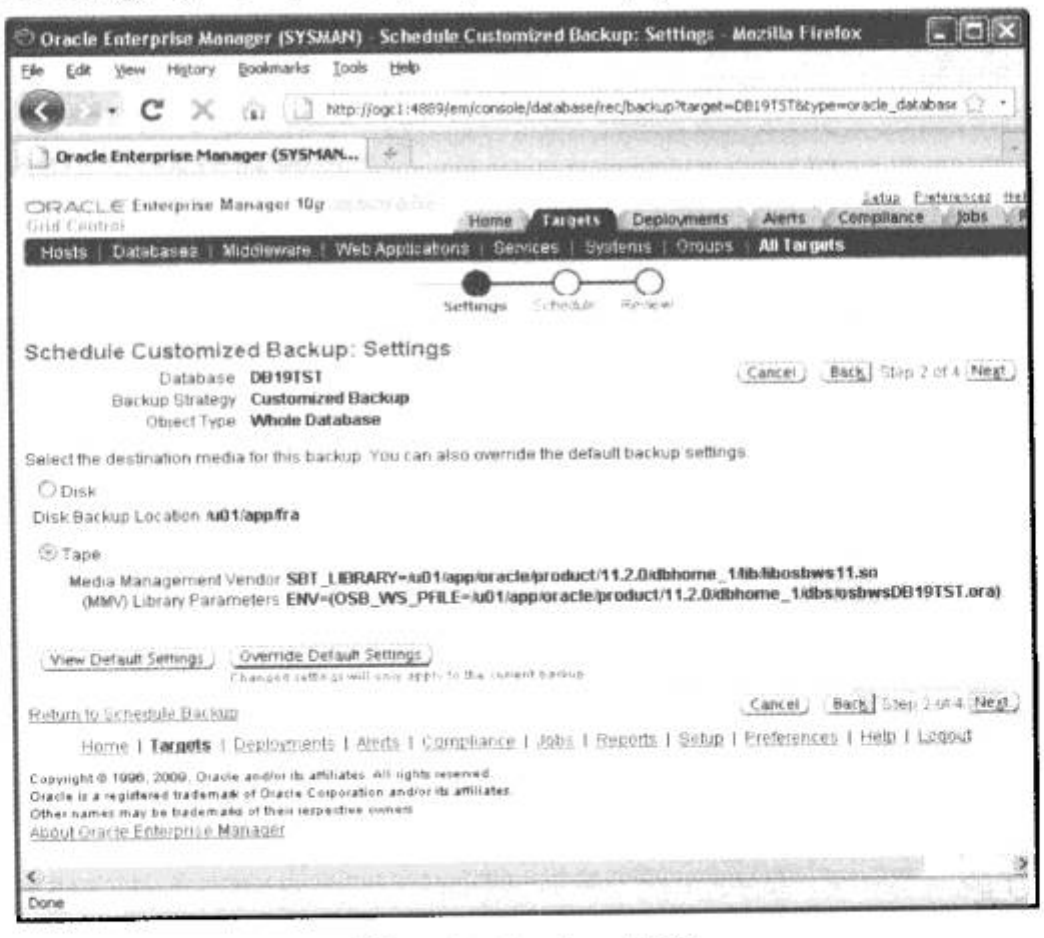


图 5-29　Settings 页面

步骤 4　选择操作类型为 one-time 类型，浏览被调度的操作，然后单击 Submit Job 按钮，如图 5-30 所示。

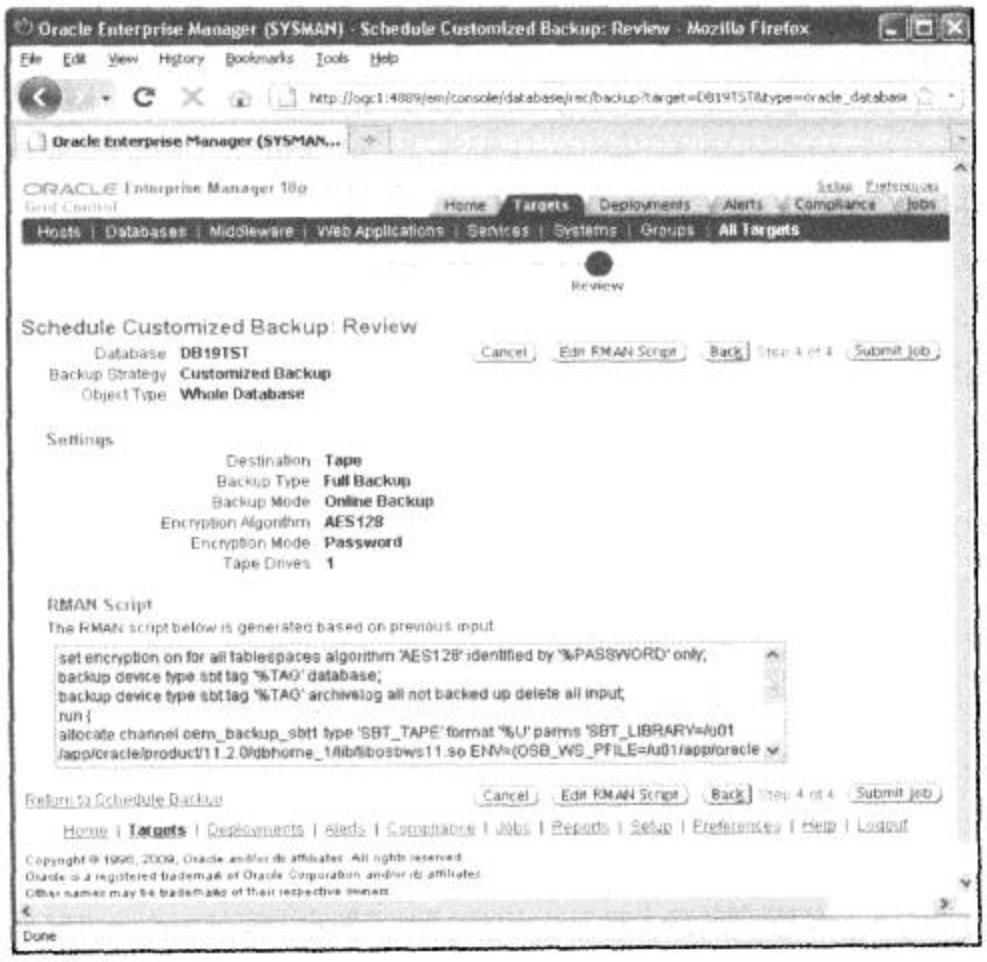


图 5-30　Review 页面

步骤 5　OEM 将显示提交的操作。执行的操作可以在 Jobs 页面中看到，如图 5-31 所示。

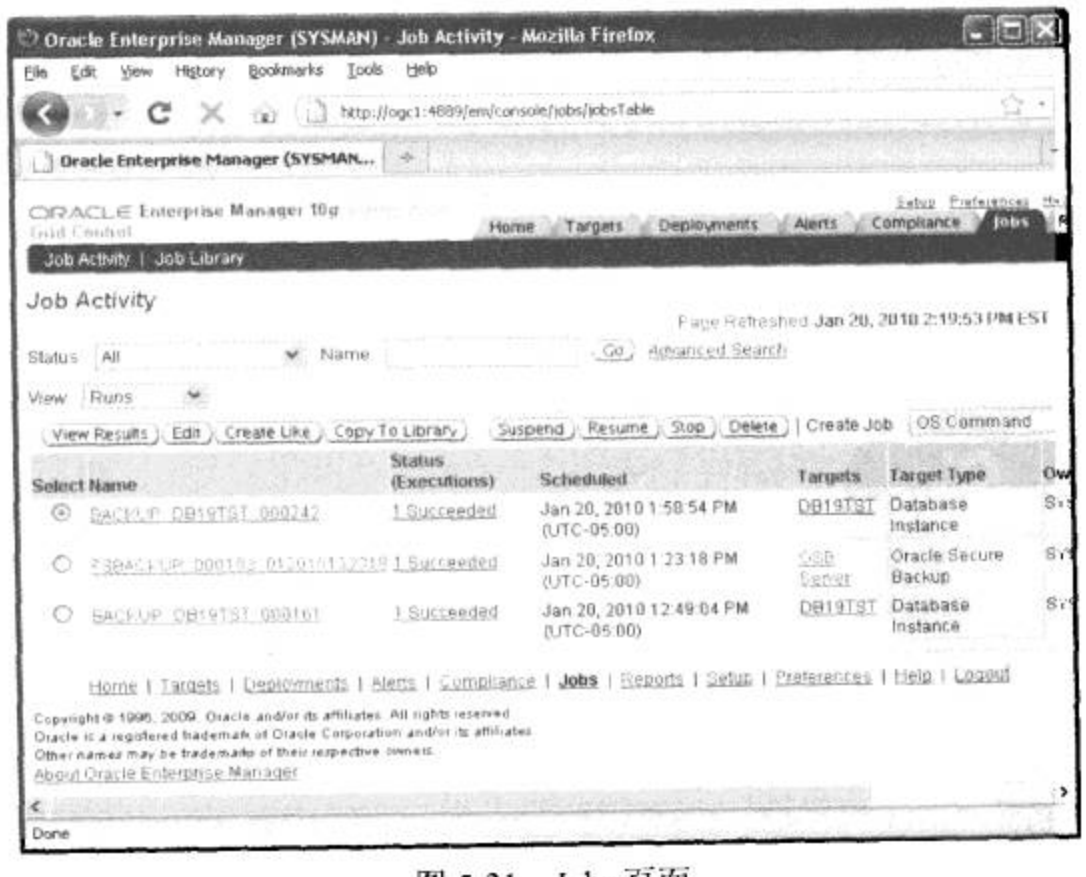
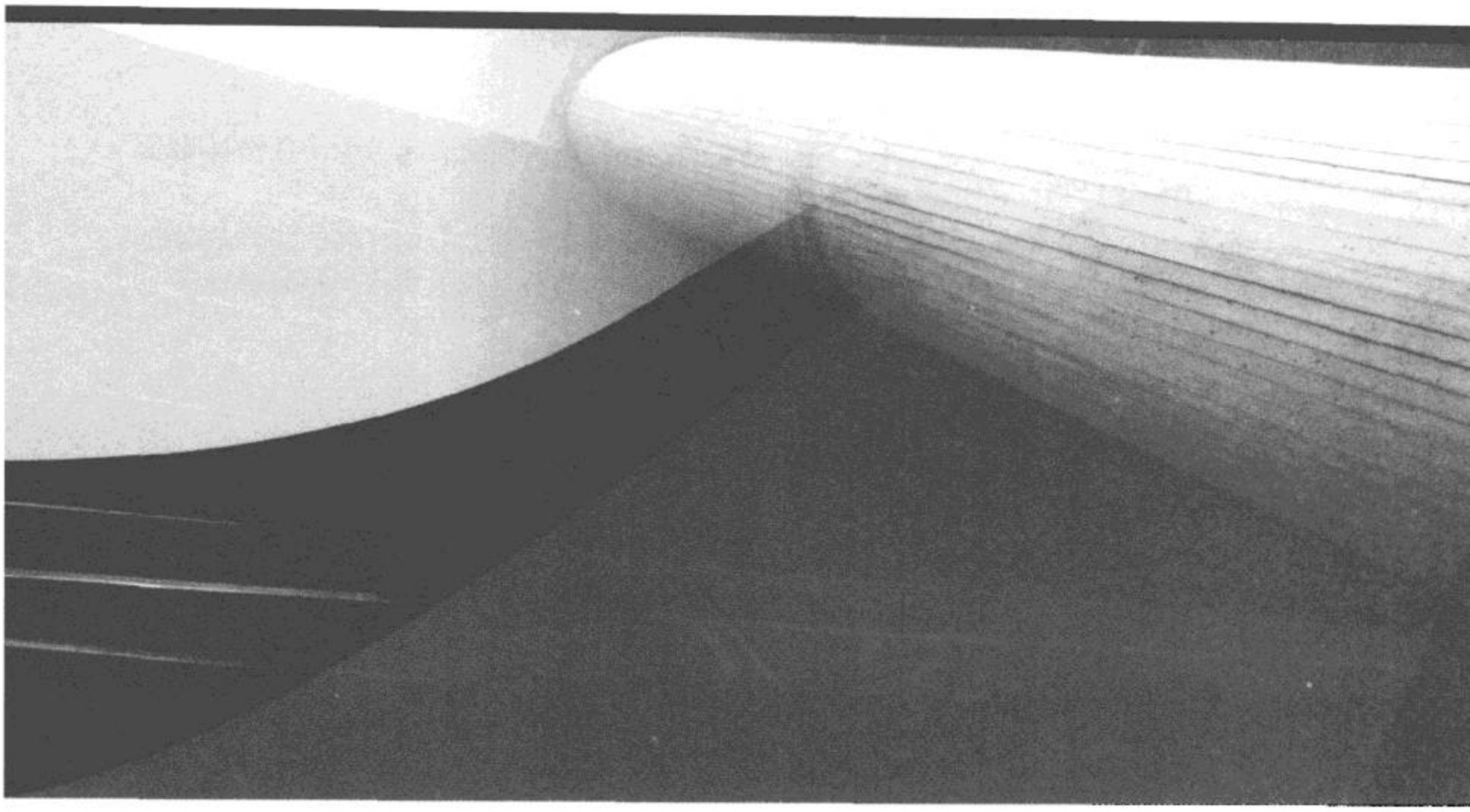


图 5-31　Jobs 页面

5.7 小结

Oracle Secure Backup 提供了高性能的、安全的数据保护，这种数据保护对于一些关键的数据的异地和本地存储都至关重要。并且来自于 Oracle Support Services 的完全产品支持、集成 Oracle 企业管理器，都具有利用云技术作为下一代异地备份存储的能力，以及非常合理的价格，这些只是使用 OSB 满足文件系统和 Oracle 数据库备份需求的部分原因。对于在为企业提供完整备份解决方案的混合的、分布式的环境中集中进行备份磁带管理，OSB 都是强有力的竞争工具。



第6章

使用 OSB 云模块将数据库备份到 Amazon Web 服务

为实现高级的 RMAN，Oracle 与 Amazon 公司共同提供了一种介质管理库，利用该介质管理库可以将 Oracle 数据库直接备份到 Amazon Web 服务云。本章主要对实现该方法涉及的重要问题进行介绍，这些问题包括：云计算，Amazon 云的工作原理，Oracle 在云计算环境下的使用，使用 Oracle OSB 云模块及 Amazon S3(Simple Storage Service，简单存储服务)执行备份对一些网站来说不失为一种好的选择。另外，本章还提供了安装和部署 OSB 云模块及 Amazon S3 作为备份解决方案的详细说明。

6.1　传统备份：前提与限制

Oracle 备份与恢复的可靠性包含两个重要的条件：可扩展的大容量现场备份基础架构和与数据库分离的位置中的定期异地备份存储。

一般来说，现场备份基础架构包含大型磁带库或磁盘阵列。异地存储的实现是通过将磁带物理转移到远程安全设备中。相关的异地磁带存储公司收取费用提供磁带转移工作和相关的远程设备。

传统方法存在以下缺点：

- 企业备份基础架构开销巨大。
- 企业备份基础架构容量受到限制。
- 异地介质转移和存储开销巨大。
- 异地备份的恢复时间长。
- 对于异地备份的物理安全性的过度依赖。

6.2　OSB 云模块

OSB 云模块是恢复管理器(RMAN)的一种介质管理库(MML，Media Management Library)，用户可以通过网络将备份直接写入到 Amazon 公司的简单存储服务(S3)中，S3 在用户看来就像是一个磁带库。而实际上 S3 是 Amazon Web 服务的一个组件，是 Amazon 的云计算平台。

云计算是一种由 Amazon 等公司大力提倡的新兴的基础架构模型，可对物理主机基础架构和异地数据存储提供可行的解决方案。云备份模型解决了多数传统备份存在的问题。此外，在云存储服务中存储备份也带来一些潜在的好处。

6.3　云计算的含义

由于大量的技术供应商通常从自己的产品和服务出发解释云计算，造成对云计算这一术语的解释在许多技术领域存在混乱的状态。

通常云计算是指通过 Internet，按用户需求的不同大小，向公众开放的存储和计算资源的远程池。某一用户的云资源与其他用户的云资源是分离和安全的。用户通过使用统一的、公开发布的软件 API(应用程序接口)管理云资源。计算云遵从供应商发布的可用性和性能服务级别的设置。

简单来说，从用户的角度来看，计算云是一种 Internet 服务，通过该服务，用户可以通过专业化管理的企业级基础架构来部署其应用程序和服务。

就 Google 来说，诸如虚拟化和映射工具等软件可以作为组件集成到用户的网页或应用程序中。

就 Amazon 公司来说，用户可以部署虚拟服务器主机和虚拟存储。从用户和应用程序的角度来看，云服务类似独立的服务器和存储设备。

6.4　Oracle 与 Amazon 云

Oracle 与 Amazon 公司合作，在 Amazon Web 服务(AWS)，也就是 Amazon 公司的云计算平台上部署了几类使用 Oracle 技术支持的组件。目前，Oracle 在 AWS 上支持以下两类服务：

- 在 Amazon AWS 弹性计算云(Elastic Compute Cloud，EC2)上运行 Oracle 软件。
- 将 Oracle 数据库备份到 Amazon AWS 简单存储服务(S3)。

6.4.1　弹性计算云(EC2)与弹性块存储(EBS)

Amazon EC2 和 EBS 允许用户部署运行 Windows 或 Linux 操作系统的虚拟主机，部署高可用的海量存储，虚拟化支持运行在这些操作系统上的所有应用程序。其中 EC2 提供虚拟主机，EBS 提供存储容量。

6.4.2　简单存储服务(S3)——Oracle 的云备份解决方案

S3 是 Amazon 云计算的另外一个组件，S3 提供低开销、可靠、冗余的海量存储服务。诸如 Jungle Disk 等流行软件包使用 S3 可以廉价地备份用户的个人计算机，或者仅获取额外的存储空间。为利用 S3 这种特性，Oracle 提供了 OSB 云模块，OSB 云模块是一种用于 S3 上的 RMAN 介质管理库。OSB 云模块允许存在于 Amazon 云上或客户网站上的 Oracle 数据库通过 RMAN 直接备份到 Amazon 云上的 S3 中。

6.4.3　RMAN 备份到 S3：OSB 云模块

OSB 云模块从本质上说是一种介质管理库，它通过 RMAN 的 SBT 信道接口提供对 Amazon S3 存储的访问。对于磁带介质管理库而言，OSB 云模块在 Linux/Unix 操作系统上作为一种共享库实现，而在 Windows 操作系统上作为 DLL(动态链接库)实现。

OSB 云模块适用于运行在 Oracle 9*i* 及以上版本的 32 位或 64 位 Linux 操作系统，32 位 Windows 操作系统和 Solaris(SPARC 64 位)操作系统上。Oracle 还为 Unix 操作系统上的任何用户请求提供端口。

6.4.4　通过 Internet 备份或 Amazon EC2 备份到 S3

驻留在 AWS 以外的主机上的数据库可以通过公共网络备份到 S3。当然这种环境下，性能可能会受到一定影响。如果数据库驻留在 AWS 上，则可以通过 Amazon 的内部网络备份到 S3，当然也能够获得良好的性能。

对驻留在 Amazon 云上的数据库，选择使用 OSB 云模块是方便的。选择 OSB 云模块可以获得廉价、可靠的备份存储。对驻留在云上的数据库来说，该方法是唯一的选择。

对于数据库驻留在 AWS 以外的情况，例如数据库在客户自己的数据中心，首先需

要考虑的是是否可以达到可接受的速度和性能。本章后面将对该问题进行讨论。

6.5 Oracle 云备份的优点

作用云备份的方法代替本地或异地磁盘或磁带存储具有一些优点，主要优点如下：

- **无需预付设备费用** 主要的资金开销在于磁带库和海量存储阵列，并且需要持续的维护和保养。在物理存储容量需要增加时，需要购买和部署新设备。与之比较，在云存储上部署备份相对廉价，即使在容量增加的情况下，也不需要增加设备。
- **低廉的持续存储开销** 按照 Q3 2009 年报告，Amazon S3 上一个月每 GB 存储开销为 15 欧元，将数据传输到服务上一个月每 GB 开销为 1 欧元。由此备份 100GB 大小的数据库外加 20GB 的归档日志，一个月的开销大约是 162 美元。
- **可伸缩性** 采用云计算方法，可以根据实际需要使用或多或少的资源。在需求增加时，不需要替换诸如磁带库等设备就可以获取更大的容量。
- **可靠性** Amazon 在其内部体系结构中提供冗余和可用性，为 S3 提供满足公认的 SLA(service-level agreement)要求的 99.99%的可用性。S3 一个关键的特性是地理复制技术。S3 将数据复制到 Amazon Web 服务领域内的三个可用性区域。可用性区域类似于独立的数据中心。如果要在用户自己的数据中心构建类似的、具有冗余和可用性的企业备份基础架构，其开销和相关工作需要大量的资金。
- **恢复时间** 不同于执行异地磁带备份，磁带必须有序存储和加载到磁带库中。RMAN 备份到 Amazon S3 总是联机完成，在恢复时随时可用。
- **无需第三方 MML 许可开销** OSB 云模块通过 Oracle 获得许可，按信道收取费用。这意味着客户可以利用他们与 Oracle 的已经存在的许可关系。

RMAN 实习：将 RMAN 备份部署到 Amazon S3 中

实习注意事项

使用 OSB 云模块的必备条件和资格:

- Oracle.com 单点登录账户(与登录到 Oracle Technology Network 的账户相同)。
- Amazon Web 服务账户。

步骤 1 如果需要 Oracle 单点登录账户，则建立该账户。如果已有 Oracle.com 或 Oracle Technology Network 账户，则可以跳过该步骤。步骤如下所示:

a. 在浏览器中，打开 http://www.oracle.com/admin/account 网页。

b. 单击 Create your Oracle account now 选项。

步骤 2 建立 Amazon Web Services 账户。

a. 在浏览器中，打开 http://aws.amazon.com 网页。

b. 单击 Sign Up Now 按钮。将有提示要求输入账户和口令登录到 Amazon 公司。如果已有 Amazon 公司的账户，则可以通过 AWS 账户和口令登录到 Amazon 公司。如果没有账户，则选择 I am a new customer 选项。

c. 登录后，必须选中复选框并单击 Continue 按钮，接受 Amazon AWS 客户协议。

步骤 3　建立好 AWS 账户后，需要注册到 Amazon 简单存储服务(S3):

a. 在浏览器中，打开 http://aws.amazon.com/s3 网页。

b. 单击 Sign up to Amazon s3 按钮。

c. 在后续网页中，需要提供用于支付的信用卡号和账单邮寄地址。最后，提示会要求浏览所做的选择并单击 Complete Sign Up。

步骤 4　需要在 S3 上存储和检索数据时，应提供个人访问标识符。注册到 Amazon S3 后，Amazon 公司会将该类标识的链接发送到您的电子邮箱中。如果您手头没有方便的 URL，则请按照下述方法执行:

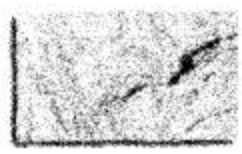
a. 在浏览器中，打开 http://aws.amazon.com 网页。

b. 将光标停留在 Your Account 上。

c. 单击 Security Credentials 按钮。

d. 注意出现的 Access Key ID 和 Secret Access Key 值。将它们安全保存，当使用账户对 AWS 服务充值时需要使用它们。

步骤 5　下载并安装 OSB 云模块安装程序。



注意:

如果您使用 Oracle 提供的一个 Amazon 机器映像(AMI)在 Amazon EC2 上执行 Oracle 数据库的备份，则不需要安装 Oracle OSB 云模块。它已经安装到目录/home/oracle/scripts/下。

可以通过 Oracle Technology Network 的云计算中心下载 OSB 云模块安装程序:

a. 在浏览器中，打开网页 http://www.oracle.com/technology/software/tech/cloud。

b. 浏览许可协议，单击 All Supported Platforms 按钮。

c. 下载文件是一个.zip 文件，该文件包含一个.jar 文件和一个 readme 文件。将.zip 文件放入将要备份的数据库服务器中能够在运行 Oracle 软件(一般放在 Oracle 中)的用户下备份。

d. 确认在服务器上安装有 java 1.5。如果没有安装，请下载并安装。注意 Oracle 11g 在$ORACLE_HOME/jdk/bin/目录中包含有 java 1.5。

```
$ java -version
java version "1.5.0_16"
```

e. 运行安装程序安装 OSB 云模块时需要提供适合于环境的参数。需要将这些参数传递给安装程序，如表 6-1 所示。

表 6-1 不同环境的参数描述

参 数	描 述
-awsid	Amazon Web 服务访问密钥 ID
-awskey	Amazon Web 服务保密访问密钥
-otnuser	Oracle.com 单点登录账号登录
-otnpass	Oracle.com 单点登录口令
-walletdir	存储先前登录凭证的目录

Linux 示例

```
$ java -jar osbws_install.jar \
> -awsid 3HKFEHKTI6JW4R88GB72 -awskey
KjsopcHzhVMAoXpPs8d0jPCa4hspbrHbssRbspbq \
> -otnuser larry_e@gmail.com -otnpass iluvb0at$ \
> -walletdir $ORACLE_HOME/dbs/osbws_wallet -libdir $ORACLE_HOME/lib
OTN userid is valid.
AWS credentials are valid.
Creating new registration for this S3 user.
Created new log bucket.
Registration ID: 0f0a8aac-dad0-6254-7d70-be4ac4f112c4
S3 Logging Bucket: oracle-log-jane-doe-1
Create credential oracle.security.client.connect_string1
OSB web-services wallet created in directory
/u01/oracle/product/11.1/db_1/dbs/osbws_wallet.
OSB web-services initialization file
/u01/oracle/product/11.1/db_1/dbs/osbwst1.ora created.
Downloading OSB Web Services Software Library.
Downloaded 13165919 bytes in 204 seconds. Transfer rate was 64538 bytes/second.
Download complete.
Extracted file /u01/oracle/product/11.1/db_1/lib/libosbws11.so
```

6.6 使用 OSB 云模块执行备份

将 RMAN 备份到 Amazon S3 的操作与将其备份到传统的 MML(介质管理库)产品中的操作类似。从 RMAN 内部看，主要有以下操作：

- 为 SBT 类型分配一个或多个信道。
- 向 SBT 信道发出备份命令。

在将 SBT 信道分配给 S3 时，RMAN 必须指定 S3 SBT 库和 OSB 云模块参数文件的位置(在安装期间创建)。通过使用 RMAN configure 命令为数据库永久定义该信息，也可以在每次分配信道时定义。如果还使用其他的介质管理库，例如 NetWorker、NetBackup、或者 Oracle Secure Backup 等，就不应该将 S3 永久配置为默认配置除非想执行该操作。

永久存储 S3 作为默认的 SBT 信道配置

使用下列代码段可以将 RMAN 的 SBT I/O 信道永久配置为默认使用 OSB 云模块的 Amazon S3。使用该方法后，SBT I/O 信道的默认目标为 Amazon S3。当其他 RMAN 备份操作需要读写本地磁带时，则需要使用 allocate channel 命令定义其本地磁带 MML 库。

```
RMAN> configure channel  device type sbt
2> parms='ENV=(OSB_WS_PFILE=/u01/oracle/product/11.1/db_1/dbs/osbwst1
.ora),
3> SBT_LIBRARY=/u01/oracle/product/11.1/db_1/lib/libosbws11.so';
```

每次分配信道时指定 OSB 云模块

下列 RMAN 代码段说明如何在脚本运行期间使用 OSB 云模块手动配置信道。SBT 信道默认配置将会保持不变。如果存在本地磁带 MML，使用该服务的操作不需要修改，将继续起作用。

```
RMAN> allocated channel  t1 type sbt
2> parms='ENV=(OSB_WS_PFILE=/u01/oracle/product/11.1/db_1/dbs/osbwst1
.ora),
3> SBT_LIBRARY=/u01/oracle/product/11.1/db_1/lib/libosbws11.so';
```

下面的 RMAN 脚本说明如何将 SBT I/O 信道手动分配给使用 OSB 云模块的 S3，并实现对数据库，包括归档重做日志的完整备份。

```
RMAN> run {
2> allocate channel t1 type sbt
3> parms='ENV=(OSB_WS_PFILE=/u01/oracle/product/11.1/db_1/dbs/osbwst1
.ora),
4> SBT_LIBRARY=/u01/oracle/product/11.1/db_1/lib/libosbws11.so';
5> backup database plus archivelog;
6> }
allocated channel: t1
channel t1: SID=133 device type=SBT_TAPE
channel t1: Oracle Secure Backup Web Services Library
Starting backup at 27-OCT-09
channel t1: starting full datafile backup set
channel t1: specifying datafile(s) in backup set
input datafile file number=00002 name=+DG1/t1/datafile/sysaux.257.700422031
input datafile file number=00001 name=+DG1/t1/datafile/system.256.700422031
input datafile file number=00003 name=+DG1/t1/datafile/undotbs1.258.700422031
input datafile file number=00004 name=+DG1/t1/datafile/users.259.700422031
channel t1: starting piece 1 at 27-OCT-09
channel t1: finished piece 1 at 27-OCT-09
piece handle=03ksrndv_1_1 tag=TAG20091027T133142 comment=API Version 2.0,MMS
Version 2.0.0.0
channel t1: backup set complete, elapsed time: 00:02:35
...
Finished backup at 27-OCT-09
released channel: t1
```

为 S3 分配 SBT 信道的语法在不同的 Oracle 版本中可能会有些变化。例如，Oracle 11gR2 不接受 ENV 关键字。在此版本下，需要使用 SBT_PARMS 关键字的信道分配语法。一些版本可能需要在 ORACLE_HOME/lib 目录中创建指向 libosbws11.so 的名为 libobk.so 的符号链接。

```
$ ln -s $ORACLE_HOME/lib/libosbws11.so  $ORACLE_HOME/lib/libobk.so
```

6.6.1 列出存储在 S3 中的 RMAN 备份以及备份集

与 RMAN 备份一样，需要由 RMAN 执行对存储备份的管理。如果手动删除备份集没有使用 RMAN，则在后续 RMAN 恢复时将始终认为它们仍然存在。在将备份标记为过时和清除 RMAN 备份时必须使用 RMAN。

类似地，最好也使用 RMAN 执行现存的备份的明细列表和报告。例如，下列命令将列出过去 24 小时内存储在 SBT 设备(例如使用 OSB 云模块的 S3)上的所有备份。

```
RMAN> list backup device type sbt completed after 'sysdate-1';
List of Backup Sets
===================
BS   Key  Type LV Size         Device Type Elapsed Time Completion Time
1         Full    1.17G        SBT_TAPE    00:02:34      27-OCT-09
          BP Key: 2  Status: AVAILABLE   Compressed: NO  Tag: TAG20091027T133142
          Handle: 03ksrndv_1_1 Media:
List of Datafiles in backup set 2
File LV Type Ckp SCN     Ckp Time Name
1       Full 1515937     27-OCT-09 +DG1/t1/datafile/system.256.700422031
2       Full 1515937     27-OCT-09 +DG1/t1/datafile/sysaux.257.700422031
3       Full 1515937     27-OCT-09 +DG1/t1/datafile/undotbs1.258.700422031
4       Full 1515937     27-OCT-09 +DG1/t1/datafile/users.259.700422031
```

目前已有一些第三方工具可用于浏览 S3 buckets(其 RMAN 备份存储在 OSB 云模块中)的内容。比较流行的一种工具是 Firefox 浏览器附带的称为 S3fox 的组件，可以通过使用 S3fox 组件浏览存储在 S3 中的 RMAN 的备份块。

6.6.2 通过 Internet 使用 OSB 云模块和 Amazon S3 优化备份与恢复

通过 Internet 备份与还原将 Oracle 数据库备份到 Amazon S3 上，其有效性取决于以下因素：

- 数据库大小
- 重做产生率
- Oracle 服务器与 Amazon S3 之间的 Internet 带宽
- 备份策略
- RMAN 选项
- 恢复时间的需求

根据 Q1 2010 年报告给出的典型的可观测的网络速度和性能，将没有驻留在 Amazon 云上的大型数据库系统通过 Internet 备份到 Amazon S3 可能不是好的选择。这一选择可

能需要花费太长的时间而难以完成备份，以及花费太长时间而不能在恢复时完成还原。同样，对于那些每天产生大量重做的数据库，选择这样的备份策略不是好的选择。Amazon Web 服务与 S3 有可靠的高速连接，因此驻留在 Amazon EC2 上的数据库不需要太多的调整和优化就可以使用 OSB 云模块。

例如，采用 0 级备份的数据库的容量为 500GB，每天产生大约 50GB 的重做可以压缩到 250GB。当采用单个 T1 线路时，备份到 S3 可能需要超过 50 小时，将使用大多可用的网络容量。显而易见，在打算通过 Internet 使用基于 S3 的备份策略时，需要仔细考虑和测试网络吞吐量，并压缩备份尺寸。需要使在备份周期里的最长的备份窗口能实现 0 级备份，同时企业能够接受其占用网络资源的比例。另外还需要确定，企业能否接受使用 S3 完成完整还原操作所需要的时间。

由于通过 Internet 将数据库备份部署到 S3 存在许多问题需要考虑，任何打算使用该方法的企业，在着手将海量数据迁移到 S3 体系结构前，都应该严格测试备份和恢复的性能。

无论数据库容量多大，使用 OSB 云模块的用户都可以使用多种方法优化备份容量和性能。通过提高备份效率，大型的数据库通过 Internet 备份到 S3 将变得可行。主要的方法包括：

- 使用多个 SBT 信道
- 在最低 Internet 使用期间备份
- 使用压缩备份集
- 采用增量备份策略
- 频繁备份归档日志

考虑使用通过 Internet 将 Oracle 数据库备份到 Amazon S3 时，需要有足够的网络带宽，保证在可接受的备份窗口内完成备份数据库的操作。如果具有足够的带宽，则问题将变为如何配置 RMAN 以使用这些资源。一种最大化利用网络资源的有效方法是打开多个 SBT 信道。测试的目标是尽量充分利用网络资源，在需要的时间范围内完成备份操作，同时将可用资源用于其他服务和意图。

为最小化对其他用户和服务的影响，将 Oracle 备份到 Amazon S3 应该在网络使用率最低的时间段进行。通过压缩备份集可以大大减少传输数据量。

增量备份策略可大大减少日常需要备份到 S3 的备份容量。在大多数数据库中每天、每周甚至每月仅有部分数据更新，因此，不需要经常使用完整备份和 0 级备份方式，增量备份可以扩展到整个晚上或周末期间进行。基于夜间或周末备份，差异备份和增量备份既可保持可恢复性，同时又使得备份容量变小。

多个信道及压缩备份集示例

以下 RMAN 脚本给出了为使用 OSB 云模块的 Amazon S3 打开多个信道的命令语法，然后通过多个信道执行包括归档重做日志在内的完全(0 级)压缩备份。

```
RMAN> run {
2>  allocate channel t1 type sbt
```

```
    3>  parms='ENV=(OSB_WS_PFILE=/u01/oracle/product/11.1/db_1/dbs/osbwst1
    .ora),
    4>  SBT_LIBRARY=/u01/oracle/product/11.1/db_1/lib/libosbws11.so';
    5>  allocate channel t2 type sbt
    6>  parms='ENV=(OSB_WS_PFILE=/u01/oracle/product/11.1/db_1/dbs/osbwst1
    .ora),
    7>  SBT_LIBRARY=/u01/oracle/product/11.1/db_1/lib/libosbws11.so';
    8>  allocate channel t3 type sbt
    9>  parms='ENV=(OSB_WS_PFILE=/u01/oracle/product/11.1/db_1/dbs/osbwst1
    .ora),
    10> SBT_LIBRARY=/u01/oracle/product/11.1/db_1/lib/libosbws11.so';
    11> backup as compressed backupset incremental level 0 database plus
archivelog;}
```

由于归档重做日志的创建具有周期性，因此可以将归档日志备份的影响分散到每天中。归档日志的频繁备份减少了在每夜数据库备份窗口期间需要备份的数据量。该方式有利于改善时间点恢复性，因为在异地可使用更多最近时间点的最近归档日志。

6.7 许可问题

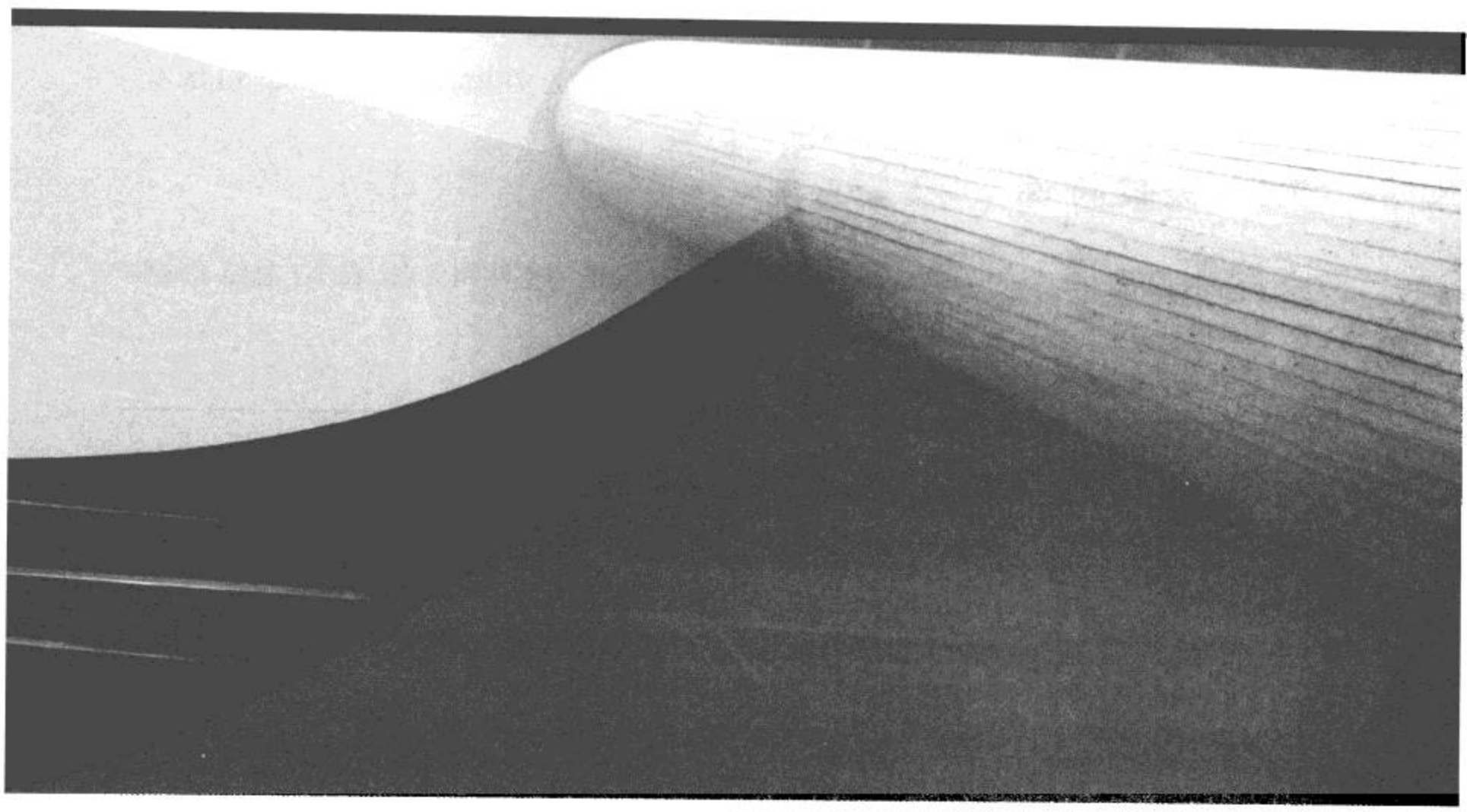
多数用于Oracle RMAN的介质管理库(MML)软件包需要第三方例如EMC或Symantec公司的许可。就OSB云模块而言，OSB许可信息文件(P/N E10310-02)声明，用户必须获得“Amazon S3目标的备份域同步使用的每个RMAN信道”的许可。

用户应详细了解该文件，所有签署的许可协议，及其法律建议，以确定正确地获取OSB云模块的许可数量。

6.8 小结

OSB云模块为安全异地备份存储和支撑提供了引人注目的解决方案，特别是当Oracle数据库驻留在Amazon云中时。与传统的磁带支撑相比，云存储具有低开销、最佳的恢复时间(TTR)及更多地理分布的灾难恢复属性的良好潜力。

当将备份部署到S3时，应该全范围地测试备份和还原的性能。此外，在考虑使用OSB云模块时，需要仔细检查Oracle包的许可责任。



第7章

使用 Oracle 的 VERITAS NetBackup 增强 RMAN

VERITAS NetBackup 服务器软件和数据库代理软件与 RMAN 协同工作来管理企业备份、恢复和存储。该产品运行在许多操作系统上，它支持流行的数据库，并且很容易与多种存储设备集成。NetBackup 具有令人着迷的特性，并且其供应商与 Oracle 密切合作使得它成为令人满意的选择。本章中的信息和下载文件可以从 Symnatec 的网站中获得，

因为 Veritas 现在是 Symnatec 的一部分。

除非特别说明，本章将主要讨论运行在 Unix 环境下的 VERITAS NetBackup 5.1 版本。

7.1 关键特性

NetBackup for Oracle 具有许多扩展了 RMAN 功能的特性和优点。表 7-1 总结了这些关键特性。

表 7-1 NetBackup for Oracle 的关键特性

特 性	优 点
备份到磁盘分级区域	通过避免磁带等待时间的开销提供较快的备份和还原
综合备份	通过允许增量备份保存网络带宽
内联副本	通过同时写入数据的多个副本提供数据冗余
磁带多路复用	通过将并行数据流从一个或多个客户端写入一个磁带驱动器来改进性能
自动化磁带设备配置	减少手动配置磁带驱动器所需的时间和工作
数据加密	通过提供多种级别的数据保护来保护写入到磁带的数据
备份模板	简化编写 RMAN 备份和恢复脚本的工作，方法是提供图形化工具来帮助生成脚本，并且提供预先配置的、和 NetBackup 一起运行的样本脚本
代理副本	在执行备份和还原操作时，将处理能力从数据库服务器转移到单独的介质服务器
检查点重新启动	允许备份从故障停止的位置开始重新执行

另外，还有一些有必要介绍的专用插件是 NetBackup Advanced Client、NetBackup Vault 和 NetBackup Bare Metal Restore。

7.2 必要的组件

下面是在 RMAN、NetBackup 服务器和存储设备之间进行成功通信交流所需的元素：

- NetBackup 服务器软件。
- NetBackup for Oracle 代理软件——包括必要的接口库文件(libobk.*)。
- Oracle 数据库软件——包括 RMAN 实用程序和 Oracle 调用接口(Oracle Call Interface，简写为 OCI)。
- NetBackup 许可——所有软件、选项和代理所需的许可。

7.3　存储/介质设备配置

通常由 Unix 管理员或存储管理员负责设置磁带驱动器、主机总线适配器、SCSI ID 和磁带机器人(参看 *NetBackup Media Manager Device Configuration Guide* 一书)。在此将不会讨论那些与特定供应商有关的步骤，而是提供一些命令来验证磁带介质设备的配置。

注意：
在安装 NetBackup for Oracle 的代理软件之前，应按照适当的工作顺序设置和测试硬件设备。

使用以下命令可从客户端服务器上查询主服务器以验证通信：

```
/<install_path>/netbackup/bin/bpclntcmd -pn
```

接下来，从客户端服务器上查询主服务器以验证版本：

```
/<install_path>/netbackup/bin/bpclntcmd -sv
```

通过执行如下命令查看哪个存储服务器将服务于客户端服务器：

```
cat /<install_path>/netbackup/bp.conf
```

最后，验证 NetBackup 通信端口监控程序是否正在监听请求：

```
netstat -a |grep bpcd
netstat -a |grep vnetd
```

7.4　NetBackup 的安装

网络化备份环境包含多个层，每一层都需要一定数量的软件配置，使得组件具有互操作能力。NetBackup 的安装过程非常直观，只需要不到 20 分钟即可完成。除了执行本地安装之外，也可能运行远程安装，从管理控制台(Administration Console)安装，以及从中心的主服务器上将软件传播给客户端。

注意：
不应该在加载网络文件系统(Network File System，NFS)的目录上安装 NetBackup 软件。这样做会干扰它的文件锁定机制。

NetBackup Server 软件安装在如下服务器上：

- 主服务器
- 介质服务器(可选)
- 客户端(数据库)服务器

NetBackup for Oracle 代理软件则安装在客户端(数据库)服务器上。

下面定义了前面提及的服务器类型：

- 主服务器：用于配合 NetBackup 环境。主服务器位于称为 Tier 1(顶层的服务器层)的层中。Tier 是根据不同体系结构层或体系结构组件的分组标记。主服务器的作用是调度备份、跟踪作业进展、管理磁带设备、以及在存储库中存储备份元数据。主服务器扮演非常关键的角色，因此从高可用性角度考虑，将该服务器群集化是很好的想法。
- 介质服务器：位于 Tier 2(中间层)中，用于在本地备份一组文件的同时跨越网络备份其他的文件。在环境中引入介质服务器是为了改进性能，但介质服务器不是必需的服务器。
- 客户端服务器：驻留在 Tier 3(客户端层)中，通常是驻留将要备份的数据库的数据库服务器。

7.4.1 安装 NetBackup for Oracle 代理之前的任务

在安装 NetBackup for Oracle 之前，需要完成如下的任务：

(1) 验证系统管理员已经在主服务器、介质服务器(可选)以及客户端数据库服务器上安装并正确配置了 NetBackup 软件。

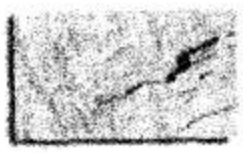
(2) 确保已经购买所有 NetBackup 服务器、客户端、代理和可选组件的许可密钥，并且已经在主服务器上进行了注册。可以在管理控制台或命令行中进行验证。在管理控制台中，执行如下命令并选择 Help | License Keys：

```
/<install_path>/netbackup/bin/jnbSA &
```

在命令行中运行如下命令：

```
/<install_path>/netbackup/bin/admincmd/get_license_key
```

(3) 获得 NetBackup for Oracle 代理软件的安装光盘或者要求 Unix 系统管理员将软件从主服务器放入客户端数据库计算机。



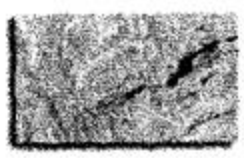
注意：
在数据库服务器上，NetBackup Server 软件和 NetBackup for Oracle 代理软件需要具有相同的版本。主服务器上的软件需要具有与数据库服务器上的软件相同或更高的版本。

7.4.2 NetBackup for Oracle 代理安装步骤

按照如下步骤安装 NetBackup for Oracle 代理：

(1) 插入并加载安装光盘。
(2) 用 root 用户账户登录到客户端(数据库)服务器。
(3) 改变加载光盘的目录。
(4) 运行./install 脚本。
(5) 选择 NetBackup Database 代理。

(6) 被询问是否希望进行本地安装。输入 y。
(7) 选择 NetBackup for Oracle。
(8) 输入 q(输入选择代理)。
(9) 输入 y 以验证前面选择的项。
(10) 安装过程如下所示。
a. 生成名为/<install_path>/netbackup/dbext/install_dbext 的脚本。
b. 使用服务器名更新文件/<install_path>/netbackup/bp.conf。
c. 在/etc/services 中添加条目。
d. 如果 NIS 在服务器上运行，则在 NIS 服务映像中添加条目。
e. 在服务器的/etc/initd.conf 文件中添加 bpcd、vopied 和 bpjava-msvc 的条目。
f. 将启动和关闭脚本复制到/etc/init.d 目录。
g. 将安装输出写入到/<install_path>/netbackup/ext。



注意：
在大多数情况下，查找安装在/usr/openv/netbackup 目录中的 NetBackup。

7.5　链接 Oracle 与 NetBackup 介质管理器的方法

NetBackup for Oracle 代理安装之后，接下来的任务是链接 Oracle 数据库软件与 NetBackup 介质管理库(MML)。该链接允许 RMAN 将文件写入到介质设备或从介质设备中取出文件。通常可以在<install_path>/netbackup/bin 中找到 NetBackup 介质管理库或 API，而 Oracle 库则位于$ORACLE_HOME/lib 中。两个文件的名称都是 libobk*。

可以自动或手动完成链接的过程，下面将对此进行介绍。

7.5.1　自动链接方法

NetBackup for Oracle 提供自动化库链接过程的脚本。因为所有步骤都是自动的，与手动操作相比，我们倾向于使用脚本。oracle_link 脚本执行如下操作：

- 检索数据库版本
- 检索操作系统版本
- 如果没有关闭数据库，则发出警告
- 检查环境参数设置
- 根据其评估应用适当的库

自动链接 Oracle 9*i* Database、Oracle Database 10*g* 和拥有 NetBackup for Oracle 的、Oracle Database 11*g* 的步骤如下：

(1) 以 Oracle 账户登录 Unix 服务器。
(2) 设置变量$ORACLE_SID 和$ORACLE_HOME。
(3) 关闭每个 Oracle 数据库实例。

```
sqlplus "/ as sysdba"
    shutdown immediate
    exit
```

(4) 运行<install_path>/netbackup/bin/oracle_link 脚本。

(5) 查看写入到/tmp/make_trace.pid 的输出以查找错误。

7.5.2 手动链接方法

如果倾向于对链接过程进行更多的控制，就可以采用手动方法。 下面是手动链接 Oracle 9*i* Database 和拥有 NetBackup for Oracle 的 Oracle Database 10*g* 的步骤：

(1) 以 oracle 账户登录 Unix 服务器。

(2) 设置变量$ORACLE_SID 和$ORACLE_HOME。

(3) 关闭每个 Oracle 数据库实例：

```
sqlplus "/ as sysdba"
    shutdown immediate
    exit
```

(4) 执行表 7-2 中合适的链接步骤。

表 7-2 手动链接过程

OS 版 本	执行的步骤	如果${ORACLE_HOME}/lib 中存在如下的文件	创建从 Oracle 库到新 NetBackup 库的符号链接
使用 64 位版本 Oracle 9*i* Release 9.0.1 和 9.2、Oracle 10*g* Release 10.1 的 AIX 64 位版本	n/a	libobk.a	mv libobk.a libobk.a.orig ln -s /<install_path>/netbackup/bin/libobk.a64 libobk.a
康柏真 64 位/数字 UNIX(OSFI)	将${ORACLE_HOME}/lib 放入 Digital UNIX 上的搜索路径中，并且设置 LD_LIBRARY_PATH	libobk.so libobk.a	mv libobk.so libobk.so.orig mv libobk.a libobk.a.orig ln -s /<install_path>/netbackup/bin/libobk.so.1 libobk.so.1 ln-s libobk.so.1 libobk.so
使用 64 位版本 Oracle 9*i* Release 9.0.1 和 9.2、Oracle 10*g* Release 10.1 的 HP-UX 64 位版本	n/a	libobk.sl libobk.a	mv libobk.sl libobk.sl.orig mv libobk.a libobk.a.orig ln -s /<install_path>/netbackup/bin/libobk.sl64 libobk.sl
Linux	将${ORACLE_HOME}/lib 放入 Digital UNIX 上的搜索路径中，并且设置 LD_LIBRARY_PATH	libobk.so	ln -s /<install_path>/netbackup/bin/libobk.so libobk.so

(续表)

OS 版本	执行的步骤	如果 ${ORACLE_HOME}/lib 中存在如下的文件	创建从 Oracle 库到新 NetBackup 库的符号链接
使用 32 位版本 Oracle 9*i* Release 9.0.1 和 9.2、Oracle 10*g* Release 10.1 的 Solaris(32 位或 64 位版本)	n/a	libobk.so	mv libobk.so libobk.so.orig ln -s /<install_path>/ netbackup/bin/libobk.so.1 libobk.so
使用 64 位版本 Oracle 9*i* Release 9.0.1 和 9.2、Oracle 10*g* Release 10.1 的 Solaris(32 位或 64 位版本)	n/a	libobk.so	mv libobk.so libobk.so.orig ln -s /<install_path>/ netbackup/bin/libobk.so64.1 libobk.so

注意：

从 Orcale 9*i* 开始，不再需要建立新的 Oracle 可执行文件。

7.6　体系结构

按照前述的步骤，到目前为止配置了硬件，安装了服务器和代理程序，运行了端口监控程序，并且链接了相应的库，已经构建了运行 RMAN 的牢固基础(参见图 7-1)。

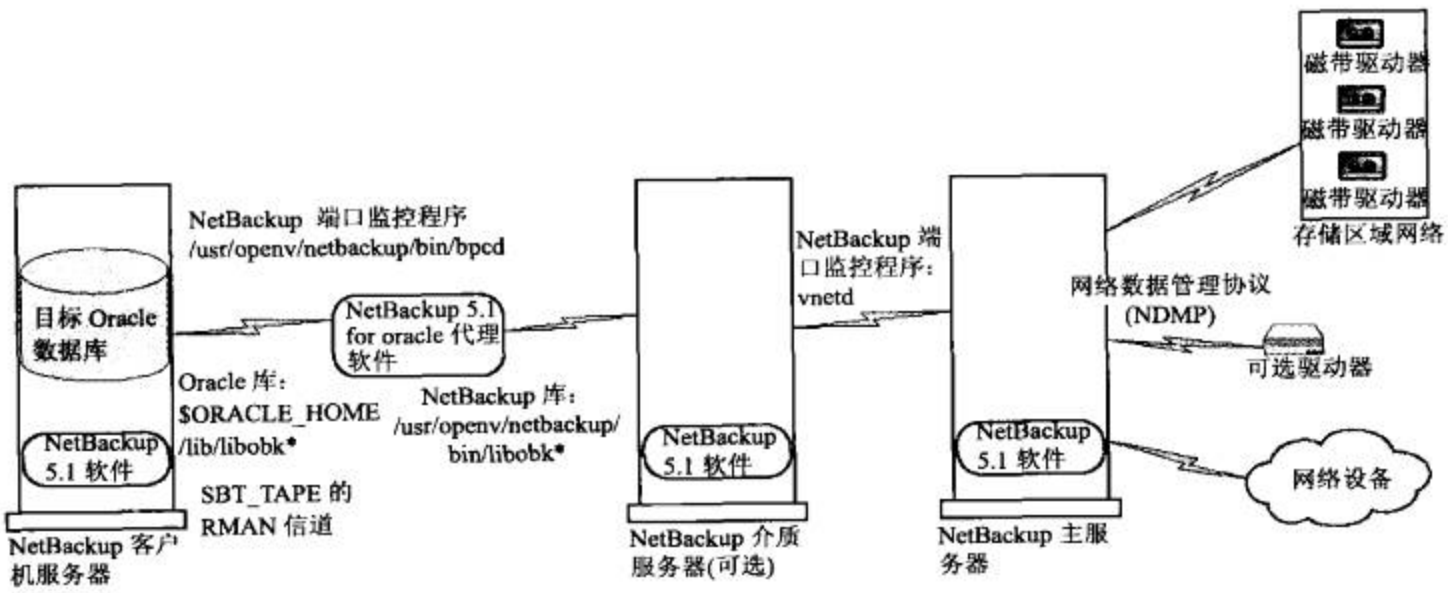


图 7-1　NetBackup 体系结构

7.7　配置 NetBackup 策略

NetBackup 需要根据指令来决定如何以及何时执行备份。这些指令被组织到称为策略的特定分组中。配置策略时需要注意以下要点：

- RMAN 作业必须至少与一个策略关联才可以执行。
- 默认的策略和代理软件一起提供。
- 可以为一个数据库服务器创建多个策略。

NetBackup 管理控制台(Administration Console)提供了用于配置如下策略信息的精美而易用的界面：

- 属性
- 时间表
- 实现策略的客户端
- 备份选项

7.7.1　添加新策略

下面是添加新策略的方法：

(1) 在将要创建策略的存储服务器上启动 NetBackup 程序。

(2) 选择 Policies 选项卡。展开 NetBackup Management | Policies，如图 7-2 所示。

(3) 在 All Policies 窗格中，右击 Master Server 选项，然后选择 New 按钮。

(4) 在 Add a New Policy 对话框中，输入一个唯一的名称，如图 7-2 所示。一旦在该对话框中添加了新的策略，就会显示 Change Policy 对话框。

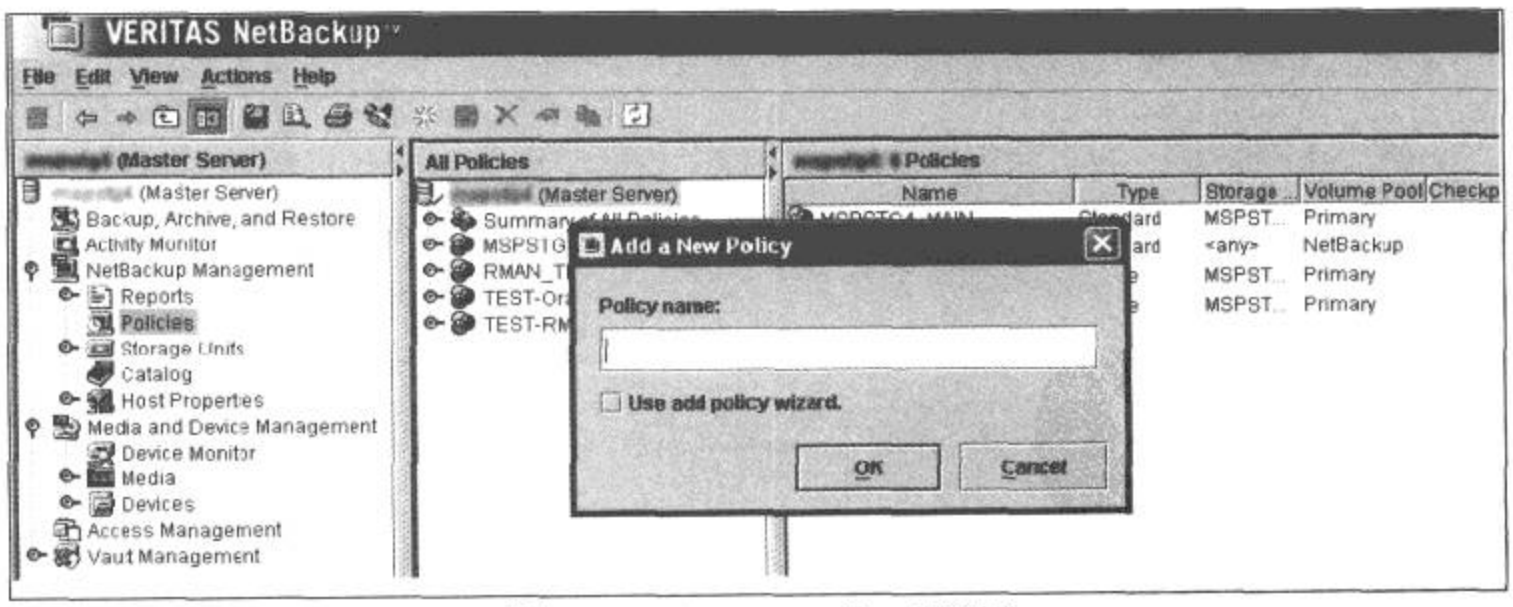


图 7-2　Add a New Policy 对话框

作为策略定义的一部分，应该定义如下的策略属性(参见图 7-3 中的属性表)。

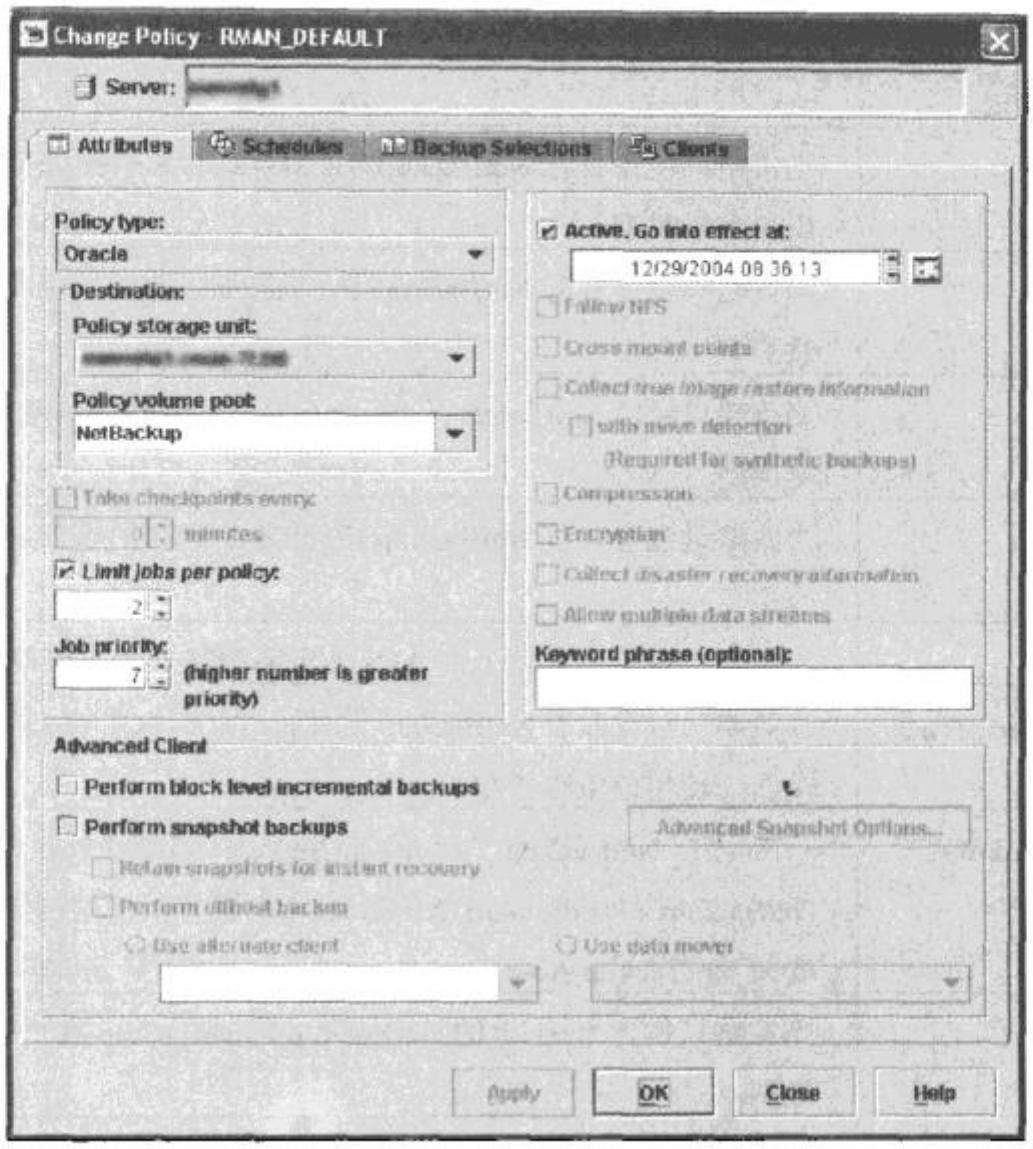


图 7-3　NetBackup 策略配置

- **策略类型**：Policy Type 下拉列表包含许多选项，对于 Oracle RMAN 备份来说可以选择 Oracle 备份类型。表 7-3 所示的是各种策略选项以及每种选项的用法。

表 7-3　策略选项及其用法

策 略 选 项	用　　法
Oracle	当策略只包含具有 NetBackup for Oracle 选项的客户端时，使用该策略类型
DB2	当策略只包含具有 NetBackup for DB2 选项的客户端时，使用该策略类型
DataStore	为 VERITAS 或其合作伙伴保留的策略类型，用于为新的应用程序或数据库提供代理
Lotus-Notes	当策略只包含具有 NetBackup for Lotus Notes 选项的客户端时，使用该策略类型
MS-Windows-NT	当策略只包含 Windows 2000、Windows NT、Windows XP 或 Windows Server 2003 客户端时，使用该策略类型
MS-Exchange-Server	当策略只包含具有 NetBackup for MS-Exchange-Server 选项的客户端时，使用该策略类型

(续表)

策略选项	用法
MS-SQL-Server	当策略只包含具有 NetBackup for MS-SQL Server 选项的客户端时，使用该策略类型
NCR-Teradata	当策略只包含具有 NetBackup for Teradata 选项的客户端时，使用该策略类型
NetWare	当策略只包含 NonTarget NetBackup Novell NetWare 客户端时，使用该策略类型
NDMP	当策略只包含具有 NetBackup for NDMP 选项的客户端时，使用该策略类型
AFS	当策略只备份客户端上的 AFS 文件系统时，使用该策略类型
DataTools-SQLBackTrack	当策略只包含具有NetBackup for DataTools-SQL-BackTrack选项的客户端时，使用该策略类型
FlashBackup-Windows	只应用于 NetBackup Enterprise Server；当策略只包含 Windows 上的 NetBackup FlashBackup-Windows 客户端时，使用该策略类型。只有在安装 NetBackup Advanced Client 时才可以使用该策略
FlashBackup	当策略只包含 Unix 上的 NetBackup FlashBackup 客户端时，使用该策略类型
Informaix-On-BAR	当策略只包含运行 NetBackup for Informix 选项的客户端时，使用该策略类型
MS-SharePoint	该策略类型用于为 NetBackup for SharePoint Portal Server 配置策略
Split-Mirror	当策略只包含具有 NetBackup for EMC 选项的客户端时，使用该策略类型
SAP	当策略只包含具有 NetBackup for SAP 选项的客户端时，使用该策略类型
Sybase	当策略只包含具有 NetBackup for Sybase 选项的客户端时，使用该策略类型
Standard	当策略只包含如下方面的任何组合时，使用该策略类型： 具有目标版本 NetBackup 软件的 NetBackup Novell NetWare 客户端 Unix 客户端(包括 Mac OS X 客户端)，除了特定策略包括的客户端(例如 Oracle)

- **备份目标**：选择 Policy Storage Unit(策略存储单元)和 Policy Volume Pool(策略卷池)的设置。
- **Limit Jobs Per Policy**：选中该复选框可以限制并行运行的作业数量。

- Active.Go Into Effect At：选中该复选框并指定一个日期和时间，从而可以在后面的日期和时间中打开预先创建的策略。

7.7.2　定义时间表

如果使用 NetBackup 调度程序，则必须定义作业何时运行。一个策略可以包含多个作业时间表，并且可以由多个数据库服务器(客户端)共享。

对于 Oracle 策略类型，存在 Application Backup Schedule(应用程序备份时间表)和 Automatic Backup Schedule(自动备份时间表)选项，下面将介绍这两个选项。根据作业的频率，将需要一个或多个自动备份时间表。

1. 配置应用程序备份时间表

当策略类型是 Oracle 时，NetBackup 就会创建应用程序备份时间表。该时间表定义了备份作业何时发生的整体时间框架。未经过调度的 Oracle 备份将默认使用该时间表。执行 RMAN 作业所需的特殊进程将作为应用程序备份时间表的一部分初始化。

配置应用程序备份时间表需要执行如下步骤：

(1) 在 Change Policy 对话框中，单击 Schedules 选项卡。

(2) 双击 Default Application Backup Schedule 选项。

(3) 单击 Attributes 选项卡，如图 7-4 所示，确保设置了 Retention(保留)设置。

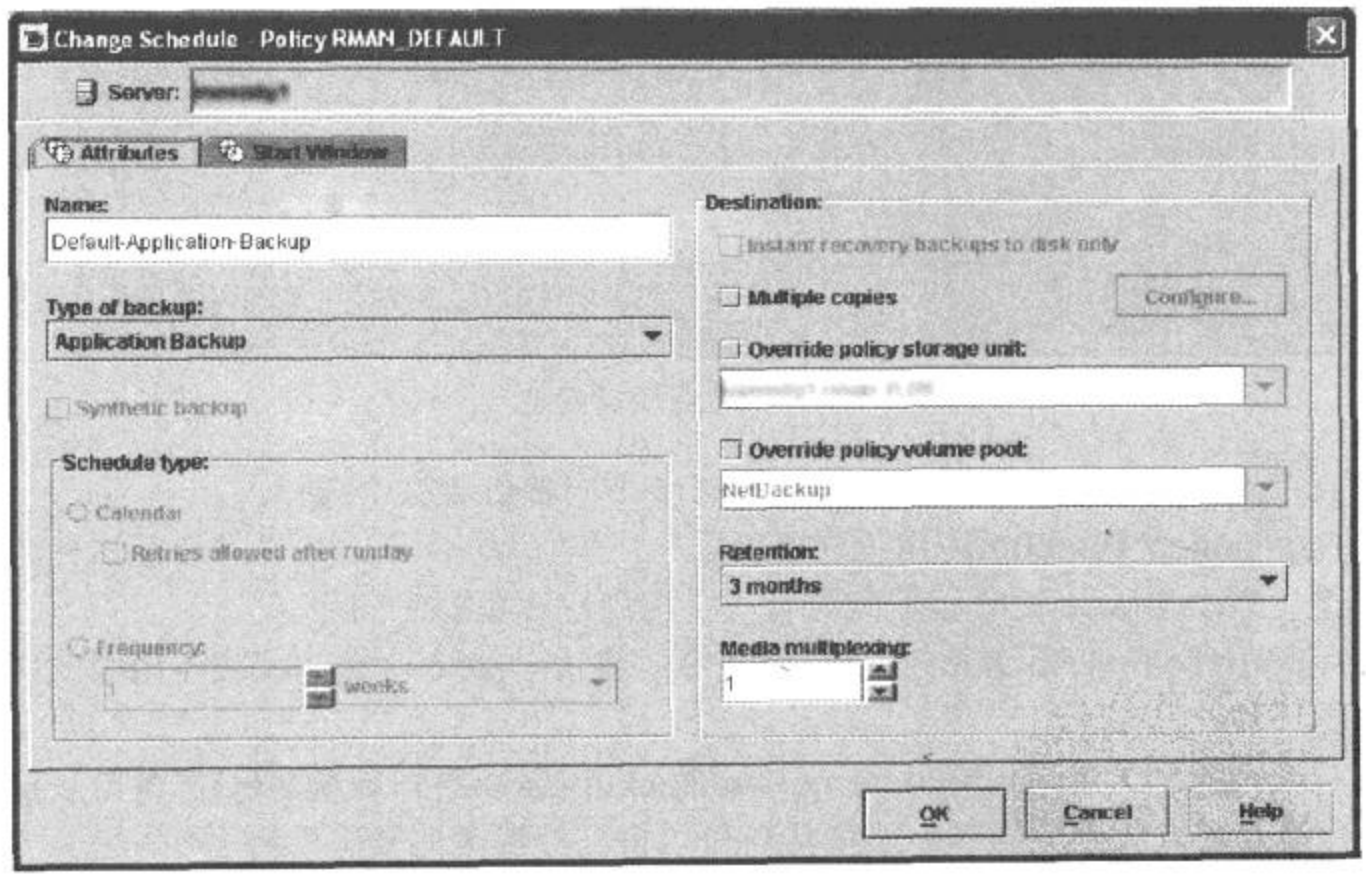


图 7-4　应用程序备份时间表

(4) 单击 Start Window 选项卡。Start Window 定义了备份作业可以开始的时间边界，这是整体的应用程序备份时间表中较为细分的子时间表。备份作业必须在 Start Window 的时间限制之内开始，但是将继续运行直到结束。

注意：
将应用程序备份时间表的备份窗口设置为一天 24 小时和一周 7 天，这样就可以在任意时间执行未调度的备份和时间表备份，并且持续运行任意时间段。

2. 配置自动备份时间表

配置自动备份时间表需要执行如下步骤：

(1) 在 Change Policy 对话框中，单击 Schedules 选项卡。

(2) 单击 New 按钮，打开 Add Schedule 窗口。

(3) 在 Attributes 选项卡上(参见图 7-5)，输入时间表的唯一名称。

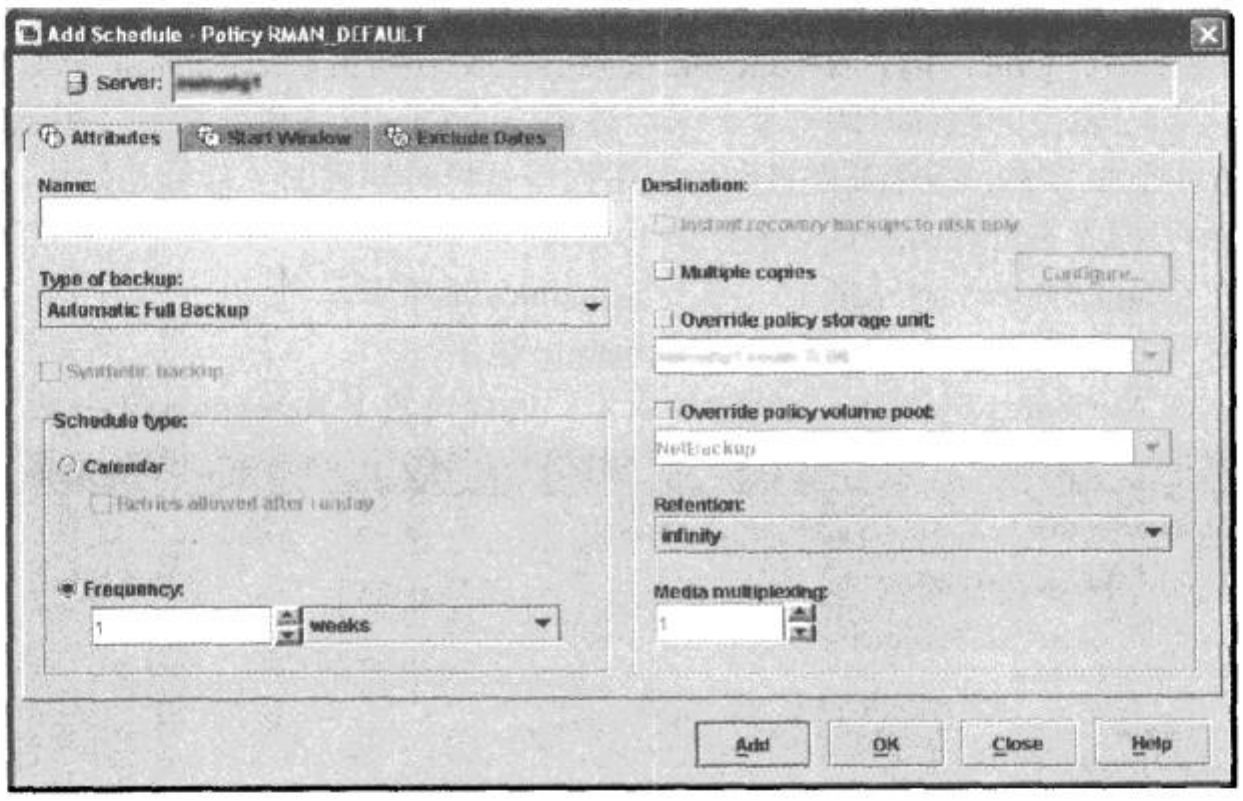


图 7-5　自动备份时间表

(4) 在 Type of Backup 下拉列表框中的 4 个不同备份类型中选择：

- Application Backup(应用程序备份)：当手动启动 Oracle 备份时运行这种备份类型。每个 Oracle 策略必须配置有一个应用程序备份时间表。
- Automatic Full Backup(自动完整备份)：备份 Oracle 已经分配或使用中的所有数据库块。
- Automatic Differential Incremental Backup(自动差异增量备份)：备份从最近的 n 级或较低级别完整备份或增量备份以来已经更改的数据库块。
- Automatic Cumulative Incremental Backup(自动累积增量备份)：备份从最近的 n-1 级或较低级别完整备份或增量备份以来已经更改的数据库块。

(5) 从两种不同的时间表类型中选择：

- Calendar(日历)：指定备份的准确日期，或一星期、一个月中的备份天数。

- **Frequency(频率)**：指定与下一个备份操作相隔的时间周期。相应选项为每小时、每天或每星期。

(6) 从 Retention 下拉列表框中选择适当的保存周期，它控制 NetBackup 保留备份记录的时间。

重复步骤(1)～步骤(6)以添加另一个时间表。

7.7.3　定义备份选项

当运行备份作业时，NetBackup 将调用任何位于备份选项列表中的自定义脚本或模板。这些文件将按照列出的顺序执行。在 NetBackup 中，有两个选项可用于定义 Oracle RMAN 备份或恢复的命令：

- **模板**：它们存储在中心主机服务器上的已知位置中，从而不需要放置到每个数据库服务器上。输入模板名时不用带路径的文件名。
- **脚本**：位于列出的每个数据库服务器上，并且必须输入完整的路径名。

按照下面的步骤向备份选项列表添加脚本或模板：

(1) 在 Administration 控制台中，双击 Policies 列表中的策略名。

(2) 在 Change Policy 对话框中，单击 Backup Selections 选项卡。

(3) 单击 New 按钮。

(4) 在如图 7-6 所示的 Add Backup Selection 对话框中，输入外壳脚本或模板名。使用 Add 按钮向选项列表添加脚本或模板。

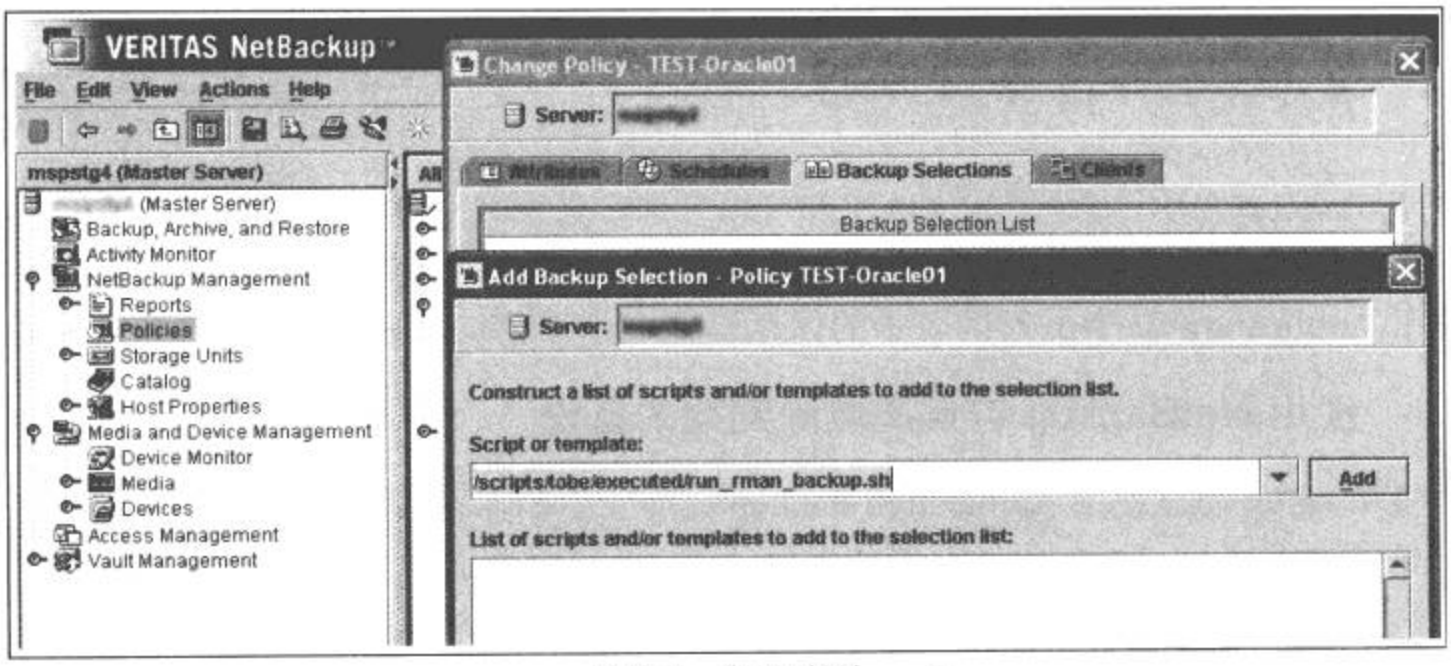


图 7-6　备份选项

(5) 单击 OK 按钮。

7.7.4　定义策略客户端

按照下面的步骤向策略添加数据库服务器(客户端)：

(1) 在 Administration 控制台中，双击 Policies 列表中的策略名。

(2) 在 Change Policy 对话框中，单击 Clients 选项卡。

(3) 单击 New 以打开 Add Client 对话框，如图 7-7 所示。

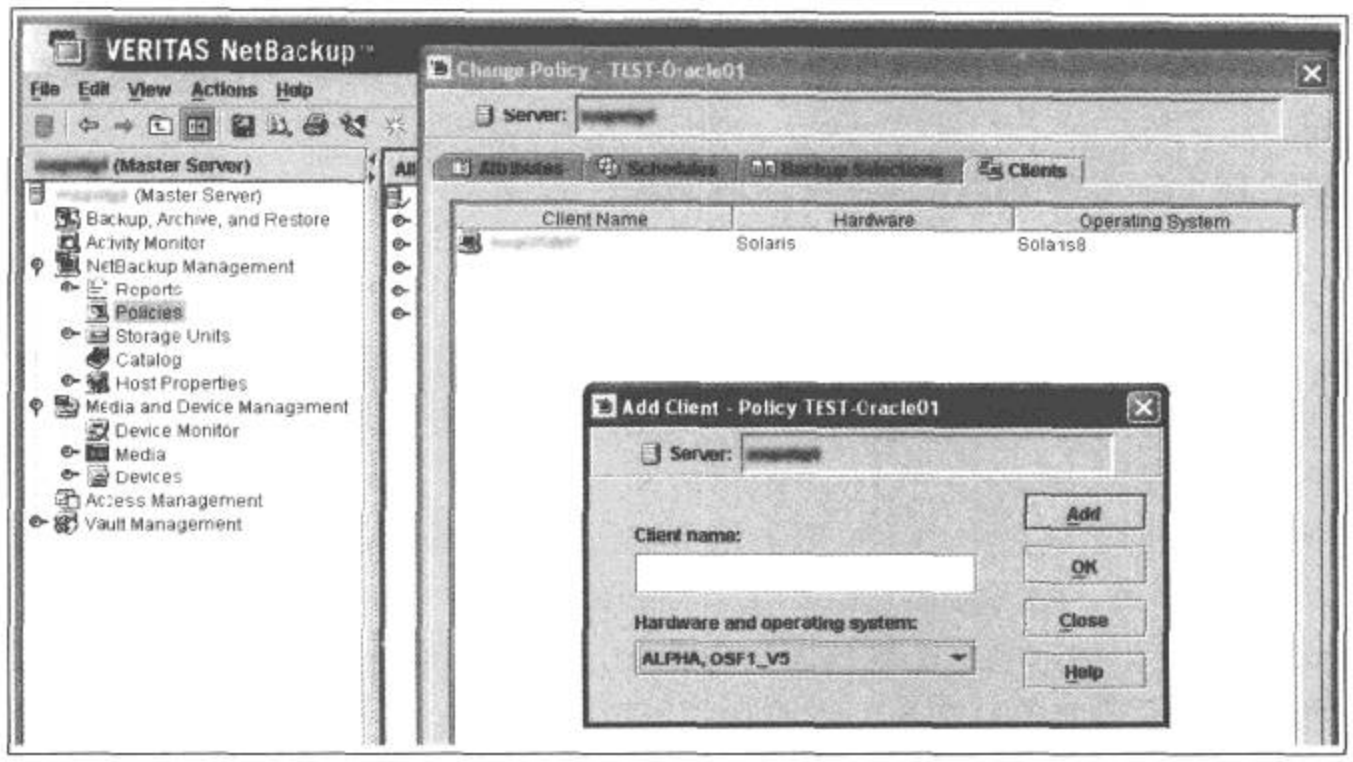


图 7-7　定义策略客户端

(4) 在 Client Name 字段中，输入添加的客户端名称。

(5) 从下拉列表框中选择硬件和操作系统类型。

(6) 单击 OK 按钮或 Add 按钮以设置另一个客户端。

7.8　管理到期的备份映像

NetBackup 介质管理器和 RMAN 都可以管理备份保存周期。如果两者的保存设置不一致，就可能产生问题。这两个存储库都不支持备份映像自动到期。解决该问题的方法是使用 Application Backup 时间表中的保存设置，然后同步 NetBackup 和 RMAN 存储库。

7.8.1　使用 NetBackup 存储库删除到期的备份

NetBackup 通过使用 Application Backup 时间表中的保存设置控制其存储库中 Oracle 备份映像的到期。这种保存设置指定备份映像到期并删除之前的时间长度。使用 NetBackup 保存设置删除备份映像时，必须执行 Oracle 存储库维护来删除对已到期备份文件的引用。

7.8.2　使用 RMAN 删除到期的备份

RMAN 提供了一个手动删除到期的所有数据库备份和归档日志备份的命令。该命令可用于删除 RMAN 目录和 NetBackup 存储库中的数据库备份。当执行请求来删除 RMAN 存储库中的备份文件时，RMAN 发送该请求给 NetBackup，从其存储库中删除相应的映像，而不管保存级别如何。删除到期的备份的代码如下所示：

```
RMAN> allocate channel for maintenance type 'SBT_TAPE';
RMAN> crosscheck backup;
RMAN> delete expired backup;
```

crosscheck 命令只在状态为“Available”的文件(实际上已被删除)可被标记为“到期”或“已删除”的情况下使用。RMAN 应该使用如下命令控制保存情况。如果使用磁带参数配置信道，则不需要分配信道，因为从 Oracle 9*i* 开始，Oracle 会自动完成这项工作。

```
RMAN> allocate channel for maintenance type 'SBT_TAPE';
RMAN> delete noprompt obsolete;
```

7.9　RMAN 样本脚本

NetBackup for Oracle 代理安装中包括 RMAN 备份和恢复样本脚本，这些脚本预先提供有(也就是说，它们已经包括代码段或模板)使用 NetBackup 的代码。可在/<install_path>/netbackup/ext/db_ext/oracle/samples/rman 中查找样本脚本。

这些样本脚本包括如下内容：

```
cold_database_backup.sh
hot_database_backup_proxy.sh
cold_duplex_database_backup_full.sh
hot_tablespace_backup.sh
database_restore.sh
hot_tablespace_backup_proxy.sh
hot_database_backup.sh
pit_database_restore.sh
```

可以通过 Administration 控制台生成新的脚本。对于曾经花费大量时间来尝试定位令人困惑的标点错误的人来说，这些脚本都会给他们带来惊喜。

下面是调用 NetBackup 的 RMAN 代码段：

```
rman target / catalog <user>/<password>@rman_cat_db log=<my_output.log>
run
{
allocate channel t1 type 'SBT_TAPE'
parms="ENV=(NB_ORA_SERV=<storage_server>,
NB_ORA_POLICY=RMAN_DEFAULT, NB_ORA_CLIENT=<db_server>)";
backup database format 'db_%d%U%t';
}
```

NetBackup Administrator's Guide 一书建议在格式字符串结尾增加%t，其原因是 NetBackup 使用时间戳作为其为目录映像搜索标准的一部分。也可以使用 configure 命令完成该工作。

下面是调用使用 configure 命令的 NetBackup 的 RMAN 代码段：

```
rman target / catalog <user>/<password>@rman_cat_db log=<my_output.log>
```

```
rman> CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'SBT_LIBRARY=/<install_path>/netbackup/bin/libobk.so64.1,
 ENV=(NB_ORA_SERV=<storage_server>, NB_ORA_POLICY=<policy_name>,
 NB_ORA_CLIENT=<db_server>)';
 rman> CONFIGURE DEVICE TYPE 'SBT_TAPE' FORMAT 'db_%d%U%t'
 rman> backup database;
```

7.10 故障诊断

在 NetBackup 环境中不可避免地会产生故障。了解如何预先区分问题的优先级可帮助建立较为平稳的问题解决过程。这一节将重点介绍诊断问题的步骤。

下面是采取的一般性故障诊断步骤：

(1) 通过确保/<install_path>/netbackup/bin 中有适当的库来验证 Oracle 代理安装。参考本章前面的表 7-2，确定哪些库(例如 libobk.a)与您的操作系统对应。

(2) 检查数据库服务器(客户端)以确保存在 bphdb 可执行文件。NetBackup 调度程序和 GUI 都使用该可执行文件启动备份。

(3) 检查是否存在如下可执行文件：

- /<install_path >/netbackup/bin/bpdbsbora
- /<install_path >/netbackup/bin/bpubsora
- /<install_path >/lib/libdbsbrman.so
- /<install_path >/lib/libnbberman.so

(4) 检查如下/<install_path >/netbackup/bin/logs 目录是否存在(具有 777 权限)：

- 在数据库服务器(客户端)上：bpdbsbora、dbclient、bphdb 和 bpcd。
- 在主服务器上：bprd 和 bpdbm。
- 在介质服务器上：bpbrm 和 bptm。

7.10.1 使用 NetBackup 日志

NetBackup 生成用于备份和还原操作的日志。这些日志可用于研究介质管理器问题，但是 RMAN 错误将写入到 RMAN 日志。下面是两种类型的 NetBackup 日志：

- **过程日志**：位于/<install_path >/netbackup/logs/user_ops/username/logs 中，生成这些日志文件是为了备份或还原时使用。这些文件有时可能非常大且繁冗，并且包含相当多的数据。注意，问题的关键是了解如何析取所需的数据。一般有两种错误类型，编号为 16 和 32；16 是错误故障(error failure)，而 32 是致命故障(critical failure)。查找这些类型错误的最佳方法是在日志文件中搜索<16>和<32>。
- **调试日志**：每个调试日志对应于一个 NetBackup 进程和可执行文件。当打开调试时，日志被写入到/<install_path >/netbackup/logs/中。这些日志可能在容量上快速增大，因此只有在必要时才使用调试。

要使得能登录数据库服务器(客户端)，需使用如下代码行修改/<install_path >/netbackup/

bp.conf 文件：

```
VERBOSE = #
```

#是 1 到 5 之间的值，用于指示日志记录级别。需要注意的是，较大的值会生成许多信息，并且可能造成目录空间用完。

注意：
确保将调试文件权限设置为 777。如果没有创建日志文件，则验证 libobk 已正确链接。

7.10.2　确定正在使用的库

以下代码用于查找哪个 NetBackup 库与 Oracle 连接：

```
ls -l $( echo $LD_LIBRARY_PATH | sed -e "s/:/ /g")/libobk* | grep libobk
```

7.11　维护安全的最佳方法

因为 NetBackup 软件运行在网络环境中，它很容易受到攻击，例如拒绝服务攻击。为了防止这些情况发生，Veritas(现在是 Symantec)推荐了如下方法。

- 只允许具有权限的用户进行管理访问。
- 只允许从受信任的服务器中进行远程访问。
- 采用最近的路径。
- 将 NetBackup 安装在防火墙后。
- 确保在服务器上运行病毒防护程序。
- 监控网络通信量以防止恶意活动。
- 阻止外部访问 NetBackup 使用的默认端口。
- NetBackup 服务器和客户端应该面向内部网络。

7.12　成本调整

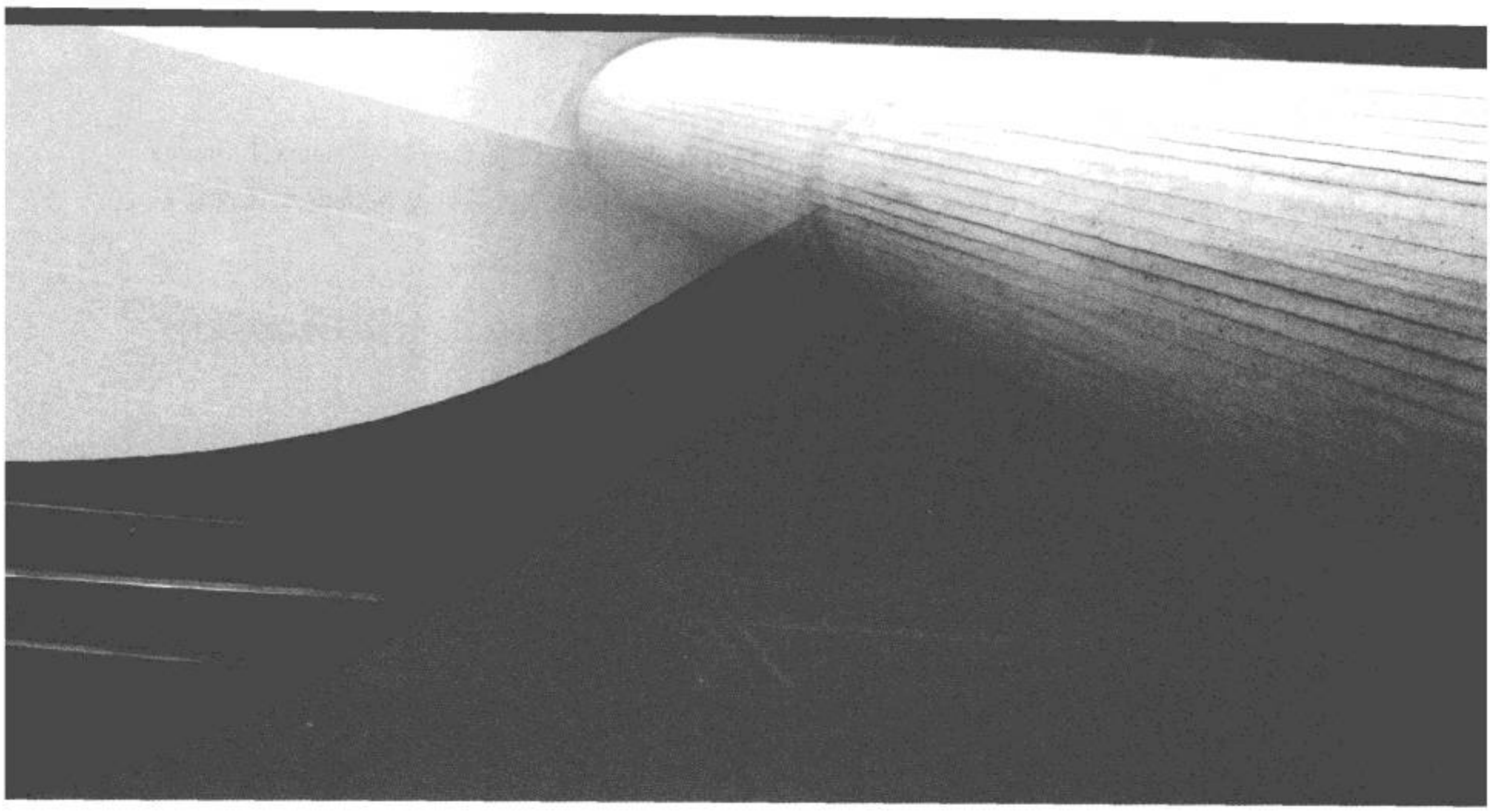
有时无法购买昂贵软件和信息技术部门的许可，这方面的成本构成了公司的无回报支出。对此，这一节将提供一些思想，说明如何管理购买 VERITAS NetBackup for Oracle 的成本。

VERITAS NetBackup for Oracle 软件扩展了 RMAN 的功能。因为该软件允许 RMAN 直接与存储服务器交流，它自动化了将由人们完成的过程。通过消除一些步骤并且删除过程中的变化情况，该软件缩短了备份和恢复时间。从本质来说，该软件可实现较好的整体应用程序性能(因为备份花费较少的时间)，减少恢复事件期间的业务停机，产生更多没有错误的恢复，并且提高数据库管理员和存储管理员的工作效率。

当面临影响工作效率和收入的严重业务停机时，NetBackup 软件就会充分体现出其价值。

7.13　小结

本章介绍了如何使用 NetBackup 软件来实现网络化的备份和恢复环境。我们概述了该软件对已有 RMAN 功能的扩展，并且介绍了如何配置直接组件通信的每一层，从而不需要手动干涉。可以发现，使用 NetBackup 增强 RMAN 会产生较快的备份和恢复，从而减少过程的变化情况，并且降低恢复事件期间的业务停机次数。NetBackup for Oracle 软件经过了细致的开发，使得部署非常简单且提供了具有丰富功能的备份和恢复解决方案。



第8章

配置 Oracle 的 HP Data Protector

在大多数环境中，对数据库管理员来说，要实现集中地调度、管理、监视和报告所有的数据库备份是一件非常困难的事情。另一个困扰数据库管理员的问题是对备份介质的管理：保护的设置，使用的监控，备份历史的检查等。然而对 HP 客户来说，在 RMAN 上使用 HP Data Protector 这样的备份用户接口将有助于数据库管理员克服上述所有的问题。

本章首先讨论 Oracle RMAN 与 HP Data Protector 之间集成的有关问题，然后描述使用 Data Protector 备份 Oracle 数据库所涉及的相关配置问题。还将介绍如何使用 Data Protector 备份和还原 Oracle 数据库。最后介绍如何设置 Oracle RMAN 元数据与 Data Protector 介质管理数据库(Media Management Database)以保持它们之间的同步。

8.1 集成 Oracle 与 Data Protector

若想成功地运行备份/还原操作，则需要采取适当的方法将 Oracle 与 Data Protector 集成。为此，需要学习支持矩阵和相关的集成组件，并通过实习，掌握集成配置操作。

8.1.1 支持矩阵

HP Data Protector A.06.00 在下列操作系统中提供对 Oracle 11*g* 恢复管理器的支持。如表 8-1 所示。

表 8-1 支持 Oracle 11*g* 恢复管理器的操作系统

Oracle 11*g* 64-bit	Oracle 11*g* 32-bit
HP-UX 11.23 (64-bit) (Itanium and PA-RISC)	SuSE Linux Enterprise Server 9 and 10
HP-UX 11.31 (64-bit) (Itanium and PA-RISC)	Red Hat Enterprise Linux 3.0, 4.*x*, 5.*x*
Solaris (Sparc) 9, 10 (64-bit) (x86 and x64)	Oracle Enterprise Linux 4.0, 5.*x* (32-bit)
Oracle Enterprise Linux 4.0, 5.0 (64-bit) (x64)	Windows Server 2003 (32-bit)
Oracle Enterprise Linux 5.3 (64-bit)	
Red Hat Enterprise Linux 3.0, 4.*x*, 5.*x* (x64)	
SuSE Linux Enterprise Server 9, 10 (x64)	
Windows Server 2003 (x64)	
Windows Server 2008 (32-bit) (64-bit)	
AIX 5.*x*, 6.1	

8.1.2 集成组件

将 Oracle 与 Data Protector 集成后，RMAN 与 Data Protector Oracle 集成软件将共同完成备份、还原、恢复、复制等操作。

Data Protector Oracle 集成代理将使用恢复目录或控制文件中的信息来确定如何执行用户提出的备份和还原操作。使用该集成，可以完成 Oracle 完整备份和增量(最高到增量级别 4)备份。Oracle 增量备份包括差异增量备份和累积增量备份两种方式。默认情况下，Data Protector 采用差异增量备份方式。如果希望指定累积增量备份方式，可以改变由 Data Protector 创建的默认 RMAN 脚本。

注意：

即使配置了 Oracle 增量备份，在 Data Protector 中，仍然不会看到定义为“incremental”的备份。Data Protector 将使用“full”代替“incremental”，原因在于在 Data Protector 中，增量备份具有另外的含义，用于为文件系统备份。

使用 Data Protector，可以执行联机和脱机数据库备份。当然，成功的备份需要适当的配置。对联机数据库备份来说，数据库实例必须处于 ARCHIVELOG 模式，对脱机数据库备份而言，数据库在备份规范中需要采用执行前和执行后备份选项。在备份执行前，可以利用 Pre-exec 选项将数据库关闭或将表空间设置为脱机状态，执行备份后，情况正好相反，通过 Post_exec 选项启动数据库或将表空间设置为联机状态。

图 8-1 显示了 Oracle 与 Data Protector 集成的体系结构。

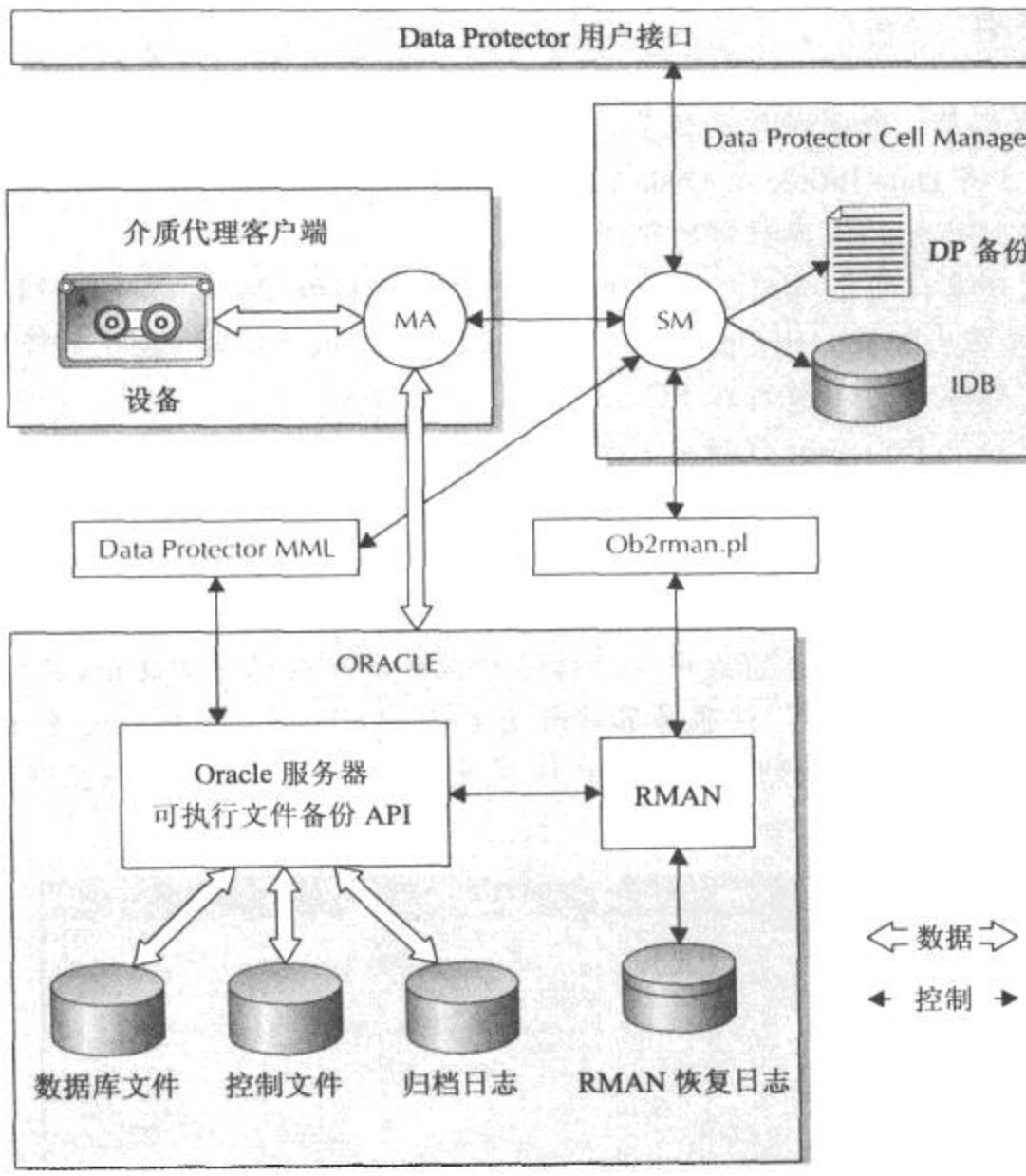


图 8-1　Oracle 与 Data Protector 的集成体系结构

在图 8-1 所示的集成体系结构中，涉及的主要组件包括：

SM　Data Protector 会话管理器，用于管理备份和还原会话。

MA　Data Protector 通用介质代理，从介质设备上或向介质设备中读写数据。

Data Protector MML　Data Protector Oracle Integration Media Management Library，该 MML 是一组例程的集合，这些例程可以确保在 Oracle 服务器和 Data Protector 之间转移数据。Data Protector MML 实现 Data Protector 与 Oracle 服务器软件之间的链接。

Ob2rman.pl　Data Protector Oracle 集成代理。与 RMAN 一起管理 Oracle 目标数据库上的备份/恢复操作所涉及的各方面的问题。

Backup API Oracle 定义的应用编程接口。

IDB 内部数据库。所有关于 Data Protector 会话的信息被写入该数据库中，这些信息包括会话消息，对象、数据、使用的设备和介质等。

RMAN Oracle 恢复管理器。

RMAN 实习：集成配置

实习注意事项

为使用 Data Protector 集成为 Oracle 数据库执行成功的 RMAN 备份，必须加载或打开 Oracle 目标数据库。如果使用了恢复目录数据库就配置它并打开，适当配置好 Oracle Net Service，并且将 Data Protector Disk Agent(磁盘代理)、Media Agent(介质代理)、Oracle Intergration 安装到目标数据库驻留的服务器上。

本实习假定所有设备和介质已配置并准备好作用，Data Protector Cell Manager(单元管理器)已安装并适当配置。HP OpenView Storage Data Protector Manager 软件将用于实现集成配置操作，该软件可以驻留在 PC 上。

步骤 1 将 Data Protector 代理安装到目标服务器上。

a. 运行 HP OpenView Storage Data Protector Manager 软件并连接 Cell Manager。

b. 在 Context List 中选择 Clients 选项，在 Scoping 窗格上右击 Clients 选项并单击 Add Clients 按钮。

c. 在 Add Client Systems 窗口中，选择目标服务器所处的平台(Unix 或 Windows 系统)，选择安装服务器，该服务器将作为 Cell Manager。单击 Next 按钮。

d. 在 Name 框中输入目标服务器的 IP 地址或者输入主机名(确定该主机名能够被正确解析)，单击 Add 按钮，然后单击 Next 按钮(如图 8-2 所示)。

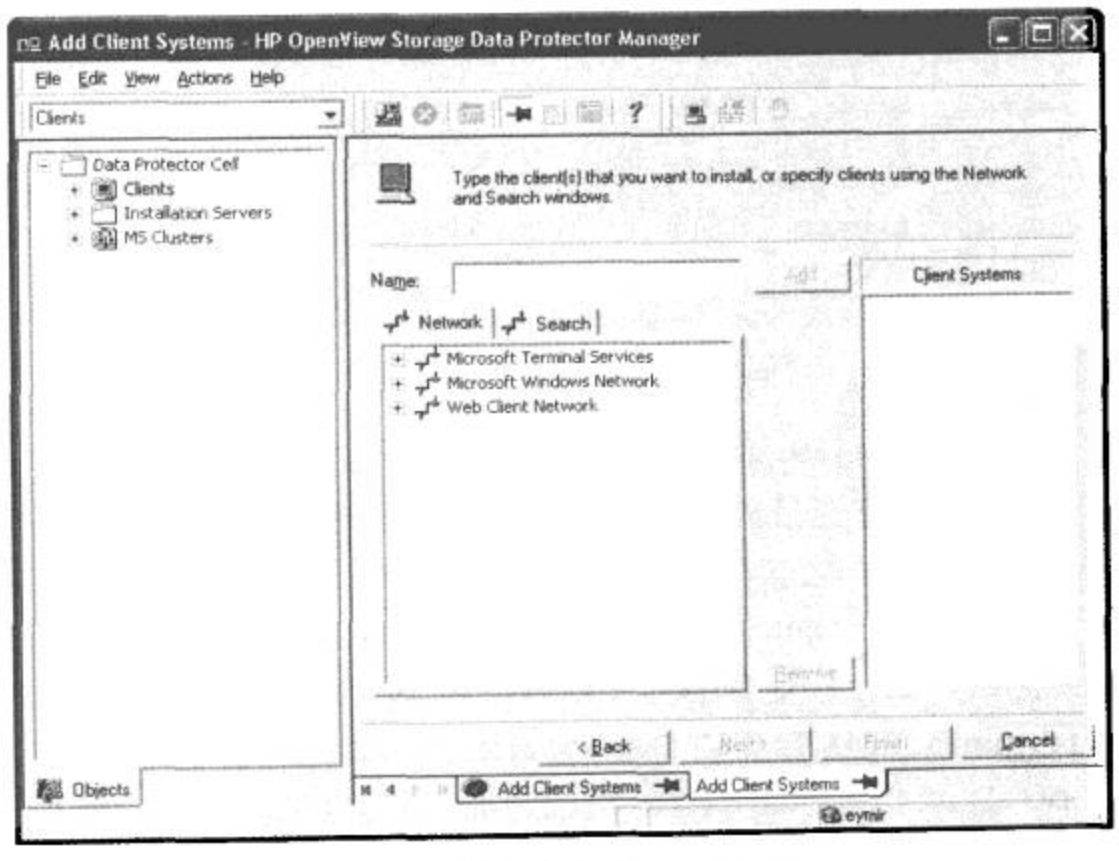


图 8-2 定义客户端 IP/主机名

e. 选择要安装的组件。为实现 Oracle 数据库备份，必须将 Disk Agent、Media Agent 和 Oracle Integration 安装到目标服务器中。选择这些组件，然后单击 Finish 按钮(如图 8-3 所示)。安装完成后，继续步骤 2 的配置。

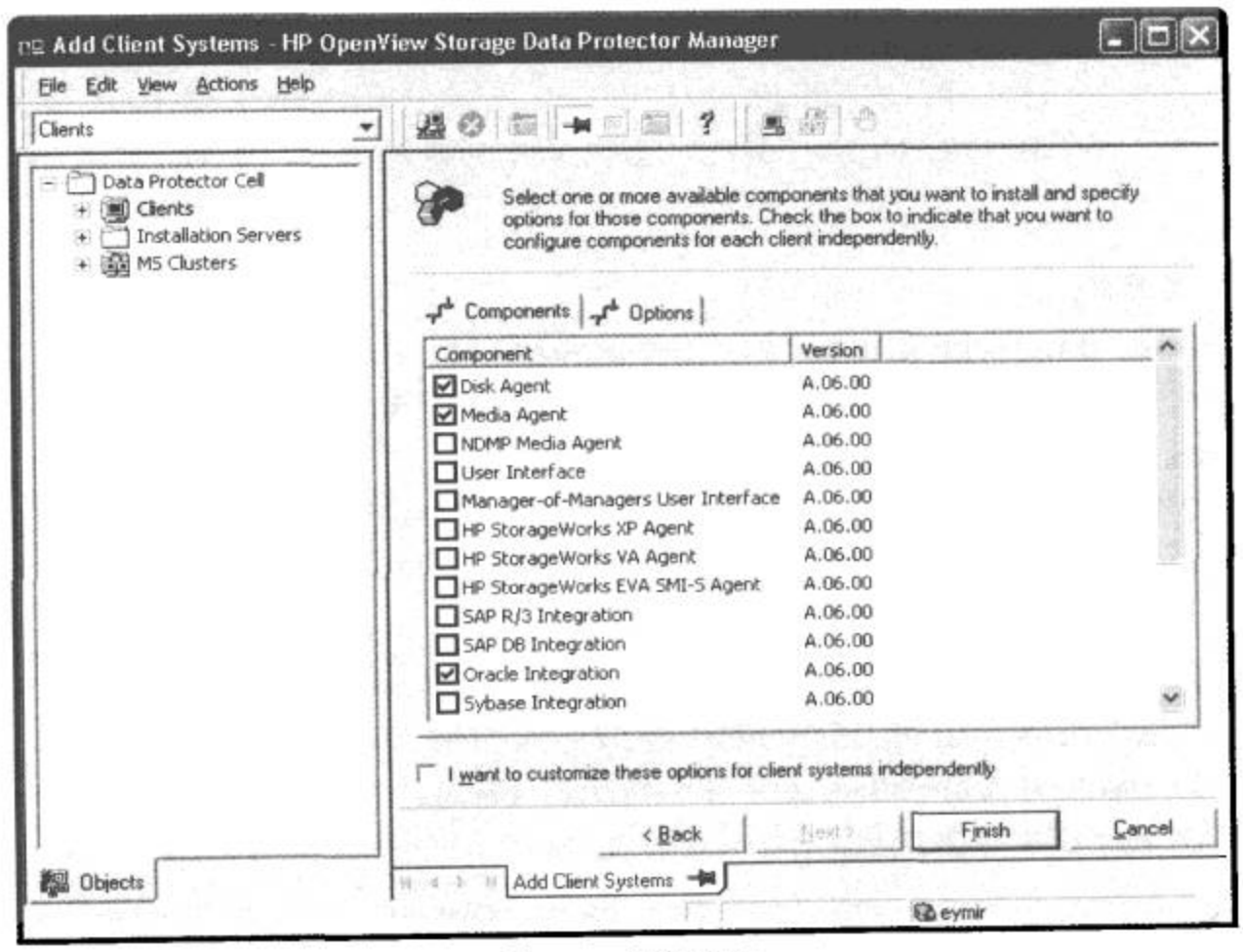


图 8-3　组件选择

步骤 2　当 Oracle 服务器需要使用 Data Protector 向(或从)设备读写数据时，该服务器将激活 MML。为使该集成正常工作，需要在目标系统上手动为 Oracle 服务器软件与 MML 建立链接。MML 位于下列目录中：

HP-UX 及 Solaris 环境　/opt/omni/lib

其他 Unix 环境　/usr/omni/lib

在不同的平台上，MML 的文件名也存在差别，如表 8-2 所示：

表 8-2　不同平台的 MML 文件名

平　台	32 位	64 位
HP-UX	libob2oracle8.sl	libob2oracle8_64bit.sl
HP-UX on IA-64	libob2oracle8.so	libob2oracle8_64bit.so
Solaris	libob2oracle8.so	libob2oracle8_64bit.so
AIX	libob2oracle8.a	libob2oracle8_64bit.a
其他 Unix	libob2oracle8.so	libob2oracle8_64bit.so

现在，可以执行如下操作：

a. 进入<ORACLE_HOME>/lib 目录。

b. 运行:

HP-UX 环境 mv libobk.sl libobk.orig

其他 Unix 环境 mv libobk.so libobk.so.orig

注意:

上述步骤仅在 libobk.sl(HP-UX 环境)或 libobk.so(其他 Unix 环境)文件已经在<ORACLE_HOME>/lib 目录中创建时执行。否则,跳过该步骤。

c. 运行

HP-UX 环境

- 32 位 In –s /opt/omni/lib/libob2oracle8.sl libobk.sl
- 64 位 In –s /opt/omni/lib/libob2oracle8_64bit.sl libobk.sl

Solaris 环境

- 32 位 In –s /opt/omni/lib/libob2oracle8.so libobk.so
- 64 位 In –s /opt/omni/lib/libob2oracle8_64bit.so libobk.so

其他 Unix 环境

- 32 位 In –s /opt/omni/lib/libob2oracle8.so libobk.so
- 64 位 In –s /opt/omni/lib/libob2oracle8_64bit.so libobk.so

若在开始备份时没有使用上述方法手动建立 Oracle 服务器软件与 MML 之间的链接,则将会出现如下的错误信息:

```
RMAN-00571: ===========================================================
RMAN-00569: ====== ERROR MESSAGE STACK FOLLOWS =======
RMAN-00571: ===========================================================
RMAN-03009: failure of allocate command on DP_TEST channel at 01/22/2010
08:54:22
ORA-19554: error allocating device, device type: SBT_TAPE, device name:
ORA-27211: Failed to load Media Management Library
Recovery Manager complete.
```

8.2 在 Data Protector 上配置 RMAN 备份

为在 Data Protector 上配置 Oracle 的 RMAN 备份,需要确定使用哪种设备、介质池,以及备份操作要使用的介质。然后可以创建 Data Protector Oracle 备份规范。

Data Protector 提供可用于创建备份规范的数据库备份模板。也可以按照需要定制模板。

RMAN 实习:备份配置

实习注意事项

按照上述的步骤,已经成功地将目标主机添加到 Data Protector 中,接下来可以定义备份规范。利用该规范,可以立即开始备份或者为备份建立时间表以在特定的时间周期

中运行。

步骤 1　运行 HP OpenView Storage Data Protector Manager 软件并连接 Cell Manager。

步骤 2　在 Context List 中，选择 Backup 选项。在 Scoping 窗格中，展开 Backup Specifications 选项，然后右击 Oracle Server 选项并单击 Add Backup 按钮。

步骤 3　在 Create New Backup 窗口中，可选择预定义的备份模板，或者若希望自定义备份操作的细节，则选择 Blank Backup 选项(如图 8-4 所示),单击 OK 按钮。

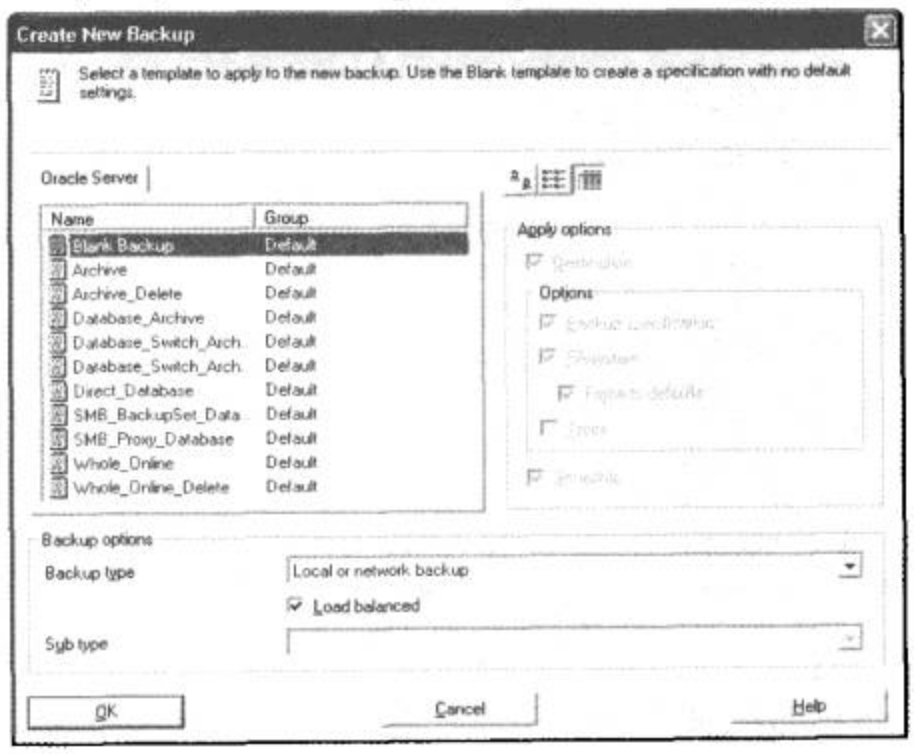


图 8-4　模板选择

步骤 4　在打开的窗口中，Data Protector 需要输入 Client(客户端)、Application Database (应用数据库)、Username(用户名)、Group Name(用户组名)等信息。指定目标数据库驻留的客户端、目标数据库的 SID、用户名以及拥有 Oracle 实例的用户组名称(如图 8-5 所示)。单击 Next 按钮。

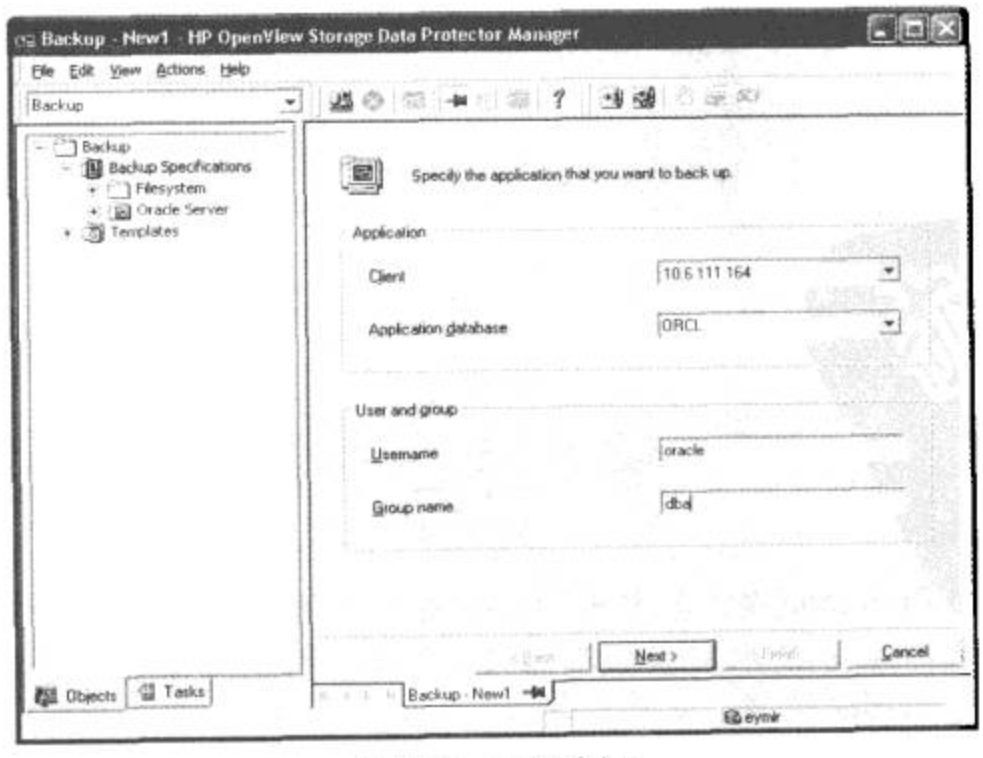


图 8-5　目标选择

步骤 5　在 Configure Oracle 窗口中，定义关于目标数据库的信息。在 General 选项卡中，定义 Oracle Server Home Directory(如图 8-6 所示)。在 Primary 选项卡中，定义用户名、口令和服务信息。记住该用户一定要授予 Oracle SYSDBA 或 Oracle SYSOPER 权限。服务是为目标数据库标识 SQL*Net 服务器进程的名称。如果使用目录数据库，则需要在 Catalog 选项卡中定义用户名、口令以及相关的服务信息。最后一个选项卡是 Standby 选项卡，若系统运行在 Oracle Data Guard 环境并且需要备份时，则需填写该选项卡。单击 OK 按钮。

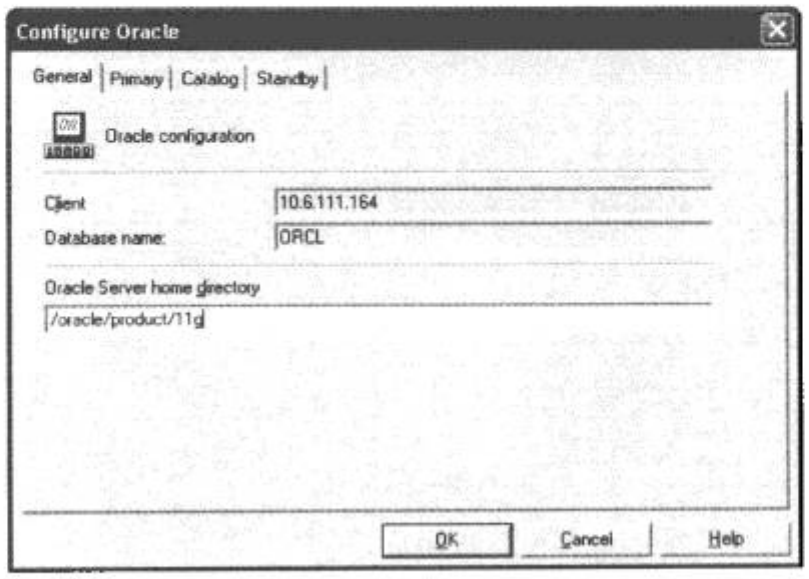


图 8-6　Oracle 配置

步骤 6　定义并说明数据库的哪些组件需要备份。如果在步骤 3 中选择 Blank Backup 选项，将看到所有未检查的组件(如图 8-7 所示)。从备选列表中选择希望备份的组件，然后单击 Next 按钮。

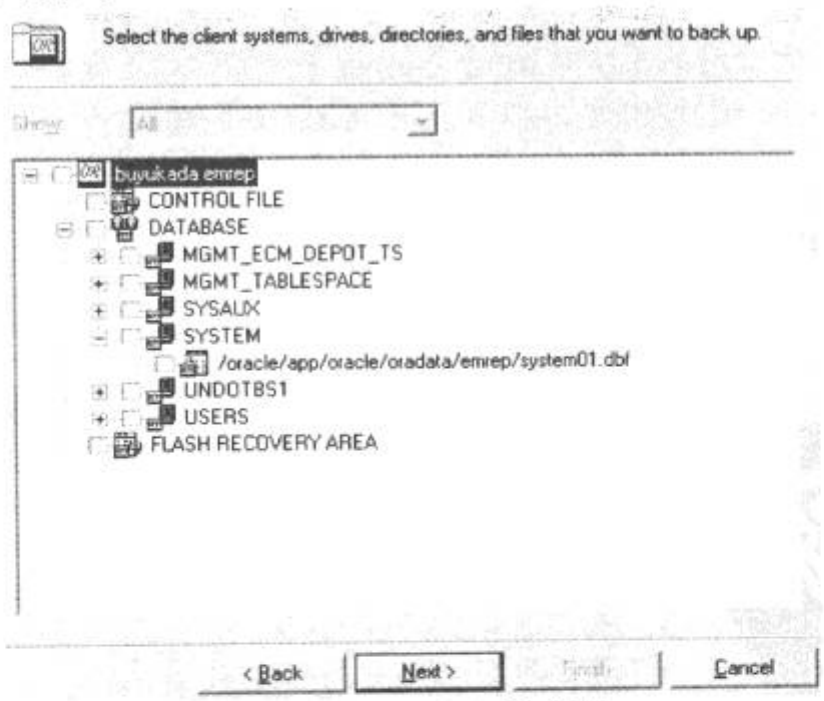


图 8-7　数据库组件选择

步骤 7　Data Protector 接下来要求确定执行该备份将使用哪些硬件。选择希望使用的驱动器(如图 8-8 所示)。若此前已经定义过介质池，则可以指定用于备份的介质池。选择驱动器，单击 Properties 选项，在出现的下拉菜单中选择介质池。选择后单击 Next 按钮。

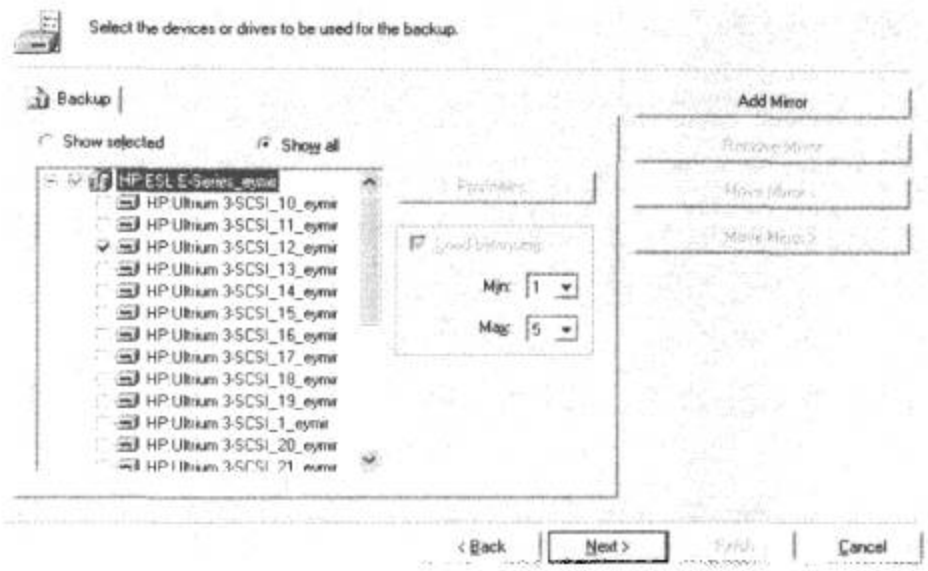


图 8-8　设备/驱动选择

步骤 8　该步骤指定详细的配置。如图 8-9 所示的三个分类中，每个都有一个 Advanced 按钮。可以在 Backup Specification Options 中定义执行前和执行后的脚本。可以在 Common Application Options 中定义备份对象的保护和报告级别。最后，可以在 Application Specific Options 中浏览已准备好的 RMAN 脚本，禁用/启用 Data Protector 管理的控制文件备份。确认做出的选择并单击 Next 按钮。

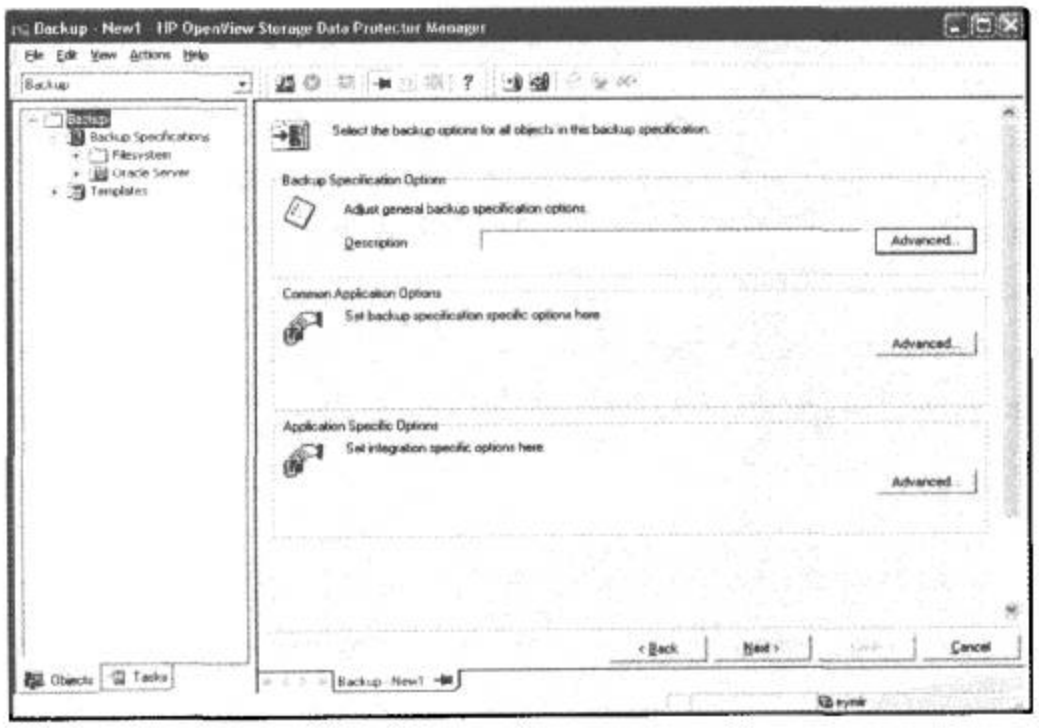


图 8-9　高级配置

步骤 9　该步骤可调度配置好的备份。指定执行备份的日期和时间。如果不希望安排好调度计划的备份工作在假日期间运行，则可以选择 Holiday 框。

步骤 10　最后一步包含以下三个选项：

- Save as 保存刚刚建立的备份/模板。
- Start Backup 采用当前制定的备份规范开始交互式备份操作。
- Start Preview 采用当前制定的备份规范开始交互式预览(测试)操作。

即使选择保存备份规范，仍然可以预览或启动备份。

编辑 Oracle RMAN 脚本

在 Data Protector Oracle 备份规范保存之后就可以编辑 RMAN 脚本部分。

若希望手动编辑 RMAN 脚本，则可以在 Context List 中，选择 Backup 选项，在 Scoping 窗格中展开 Backup Specifications，然后展开 Oracle Server，单击将要编辑的备份规范，在 Options 选项卡中，单击 Application Specific Options 框中的 Advanced 按钮，显示出的 RMAN 脚本带有一个 Edit 按钮。若单击该按钮将出现一个警告：

“该操作将立即保存备份规范，并且在 RMAN 脚本保存后将重新打开编辑器，希望继续吗？”

单击 Yes 按钮可以手动配置脚本。可以单击 Save 按钮保存该配置。

如果 RMAN 脚本包含额外的手动输入的备份命令，例如，一条备份数据库的备份命令已经存在于前一个备份命令中，对象选择将被禁用，只能浏览 Source 选项卡。

8.3 RMAN 备份

到目前为止我们已将 Oracle 与 Data Protector 集成，并且在 Data Protector 上配置了 RMAN 备份，下面将学习在 Data Protector 集成的 Oracle 数据库上运行 RMAN 备份的方法。

8.3.1 备份方法

为启动 Data Protector 集成的 Oracle 数据库的 RMAN 备份，可以采用以下的三种方法：

- 使用 Data Protector GUI(图形用户界面)或者 Data Protector CLI 启动一个预先定义的 Oracle 备份规范的交互式备份。
- 使用 Data Protector Scheduler 调度执行一个预先定义的 Oracle 备份规范的备份。
- 使用 Oracle 恢复管理器或 Oracle 企业管理器启动在 Oracle 服务器上的备份。

1. 运行交互式备份

使用 Data Protector GUI 启动 Oracle 数据库的交互式备份。

(1) 在 HP OpenView Storage Data Protector Manager(Data Protector GUI)的下拉菜单中选择 Backup 选项。

(2) 在左边的窗格中，选择 Backup | Backup Specifications | Oracle Server 命令。

(3) 右击希望使用的备份规范，并选择 Start Backup 选项。

2. 调度备份

为调度 Oracle 备份规范。选择以下步骤：

(1) 在 HP OpenView Storage Data Protector Manager 的下拉菜单中选择 Backup 选项。

(2) 在左边的窗格中，选择 Backup | Backup Specifications | Oracle Server 命令。

(3) 双击希望调度的备份规范并单击 Schedule 选项卡。

(4) 在 Schedule 页面中，从日历上选择一个日期，然后单击 Add 按钮来打开一个 Schedule Backup 对话框。

(5) 指定需要的调度选项。

3. 使用 RMAN 或企业管理器启动 Oracle 数据库备份

也可以使用 RMAN CLI 或企业管理器执行 Data Protector 集成的数据库的备份操作。为使用 Oracle 数据库备份中的 Data Protector 备份介质，需要将信道类型定义为 SBT_TAPE 类型。

8.3.2　备份过程

当使用 Data Protector 启动备份时，后台将执行下列操作：

(1) Data Protector 执行 ob2rman.pl，该程序启动客户端的 RMAN 并向客户端发送预先配置的 RMAN 脚本。

(2) RMAN 联系 Oracle 服务器，Oracle 服务器通过 MML 接口联系 Data Protector 并初始化备份。

(3) 在备份会话期间，Oracle 服务器从磁盘读取数据并将数据发送到 Data Protector，由 Data Protector 将数据写入到备份设备中。

(4) 在 Data Protector 备份会话期间产生的消息以及由 Oracle 数据库创建的消息将被记录到 Data Protector 数据库中。

8.4　使用 Data Protector GUI 还原 Oracle 数据库

可以使用 Data Protector GuI 和 RMAN 还原下列数据库对象：

- 控制文件
- 数据文件
- 表空间
- 数据库
- 恢复目录数据库

也可以通过使用 Data Protector GUI 复制数据库。要还原或复制数据库则需要创建一个 Oracle 实例。

在还原数据库项或复制数据库时，应确保数据库处于正常的状态，如表 8-3 所示。

表 8-3　数据库还原的状态

还　原　项	数据库状态
控制文件，数据库复制	非加载(已启动)
其他项	加载

注意：
在仅仅需要还原一些表空间或数据文件时，可以打开数据库，使将被还原的表空间或数据文件处于脱机状态。

在还原时，要使用适当的命令创建 RMAN 脚本，取决于 GUI 中作出的选择。若想执行额外的操作，则不能编辑 RMAN 还原脚本，但仍然能够通过 RMAN 本身手动执行操作。

8.4.1 还原控制文件

根据控制文件备份的类型，可以使用三种还原方法还原控制文件：

1. 从 Data Protector 管理的控制文件备份中还原(CONTROLFILE FROM DP MANAGED BACKUP)

在备份会话结束前，控制文件被 ob2rman.pl 自动备份，除非选择 Disable Data Protector managed control file backup 选项。

该还原选项不需要恢复目录。

控制文件被还原到下列目录：

Windows 环境　<Data_Protector_home>\tmp\ctrl<DB_NAME>.dbf

HP-UX 及 Solaris 环境　/var/opt/omni/tmp/ctrl<DB_NAME>.dbf

其他 Unix 环境　/usr/opt/omni/tmp/ctrl<DB_NAME>.dbf

还原会话完成后，必须执行下列脚本：

```
run {
allocate channel 'dev0' type disk;
restore controlfile from '<TMP_FILENAME>'
release channel 'dev0'
```

其中<TMP_FILENAME> 是文件被还原的位置。

2. 从 RMAN 自动备份中还原(CONTROLFILE FROM RMANAUTOBACKUP)

RMAN 自动备份控制文件，恢复目录不可用。

3. 从 RMAN 备份集中还原(CONTROLFILE FROM RMANBACKUPSET)

需要使用恢复目录。

8.4.2 还原 Oracle 数据库对象

为还原 Oracle 数据库对象，可以执行以下步骤：

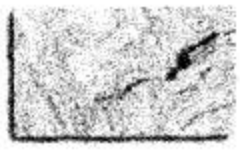
(1) 将数据库置于加载状态。

(2) 进入 Data Protector GUI，切换到 Restore 环境中。

(3) 在 Restore Objects 中，展开 Oracle Server，展开将要还原的数据库对象驻留的数据库的客户端，然后单击该数据库。

(4) 在 Restore Action 下拉列表中，选择希望执行的还原类型。

(5) 在 Results Area 中，选择还原的对象。如果要还原数据文件，则可以将文件还原到新位置。右击数据库对象，单击 Restore As 按钮，在出现的 Restore As 对话框中，指定新的数据文件位置。



注意：

在为还原指定新的位置时，只有从 Restore Action 下拉列表中选择 Perform Restore and Recovery 选项时，当前的数据文件将被切换到还原的数据文件副本上。

(6) 在 Options 页面上，从 Client 下拉列表中选择将启动 Data Protector Oracle Integration 代理的客户端。将数据库对象还原到与选择的代理不同的数据库上，单击 Settings 按钮并指定目标数据库的登录信息。

(7) 在 Devices 页面中，选择将用于还原的设备。尽管 Data Protector 默认将使用备份采用的设备作为还原的设备，但仍可以选择与备份使用的设备不同的设备用于还原。要更改还原数据库的项的设备，则选择希望使用的设备并单击 Change 按钮。

(8) 单击 Restore 按钮。

8.5 Oracle RMAN 元数据与 Data Protector 介质管理数据库同步

RMAN 元数据包含了关于目标数据库的信息，RMAN 元数据可以存储在恢复目录数据库中，也可以存储在控制文件中。RMAN 使用这些信息完成所有的备份、还原和维护操作。

Data Protector 拥有自己的数据保护策略，该策略与 Oracle RMAN 元数据不能实现自动同步。为使两个目录同步，则需要使用 RMAN 执行下述命令：

```
allocate channel for maintenance type 'sbt_tape' parms 'env=(OB2MAINTENANCE=
1)';
crosscheck backup completed after "TO_DATE('01/13/10 12:00:00','MM/DD/YY HH24:
MI:SS')";
release channel;
```

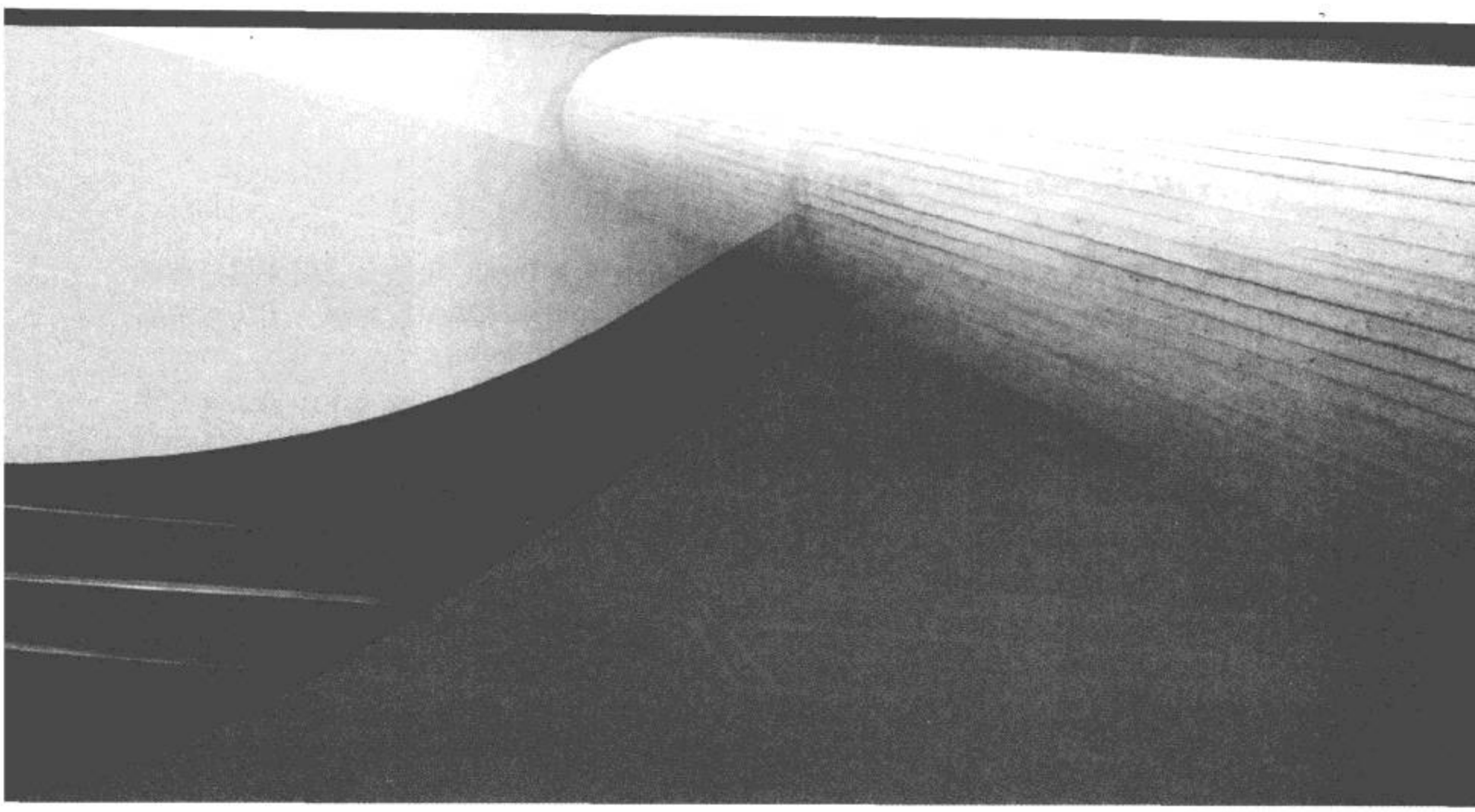
RMAN 将检查其存储库中的所有备份信息并查询 Data Protector 的内部数据库，确定备份片的可用性。随后，RMAN 根据介质的可用性将备份片标记为“过期”或“可用”。如果备份信息在 Data Protector 的内部数据库中过期，RMAN 不会将备份信息删除，而是在 RMAN 存储库中将这些信息标记为过期。

要从恢复目录数据库中删除过期的备份对象，则可以使用 RMAN 运行下列命令。

```
delete expired backup;
```

8.6 小结

本章概述了使用 HP Data Protector 软件执行 Oracle RMAN 备份的方法。在正确地完成了对该集成方法的配置后，可以按照备份需求准备和调度备份配置，使用该方法，对备份/还原操作的管理将变得非常容易。



第9章

RMAN与Tivoli Storage Manager

如果已经使用(Tivoli Storage Manager，TSM)备份企业中的文件，则进一步采用 TSM 备份 Oracle 数据库就有许多实际意义：不仅可以利用已有的数据保护资产，还可以获得从 Oracle 的 RMAN 实用程序到 TSM 的无缝连接。只需要对 RMAN 脚本进行一些简单修改，并且安装一次 TSM 客户端，甚至不用知道用于备份的磁带或磁盘驱动器位于不同的服务器上。作为 DBA，您甚至可能不需要运行 TSM 控制台命令。

本章中将介绍与 TSM 相关的一系列主题、TSM 客户端概述，以及称为(Tivoli Data Protection for Oracle，TDPO)的附带的模块。首先，我们将简要概述 TSM 体系结构以及 Oracle 客户端如何连接该体系结构。一旦开始在执行 RMAN 备份命令的服务器上测试和配置了 TDPO，就意味着您将开始深入接触 TSM。

本章也将简略地介绍一些 TSM 和 Oracle 客户端实用程序，使用这些实用程序执行初始和例程配置以及监控任务。

本章的最后将介绍采用 TSM 和 TDPO 备份 Oracle 数据库时可能会遇到的常见问题，并讨论如何解决这些问题。

9.1　Tivoli Storage Manager 概述

TSM 是多层体系结构：当使用 TSM 备份 Oracle 数据库时，可能会使用 4 层。然而您可能会将所有层驻留在单一服务器上，但在分布式环境中不推荐这样做，因为在分布式环境中需要将备份服务器与目标服务器分离。

图 9-1 是一个典型的 TSM 环境图。在下面几节中，将详细介绍图 9-1 中显示的每个组成部分，并且按照介绍的过程解释一些 TSM 概念。

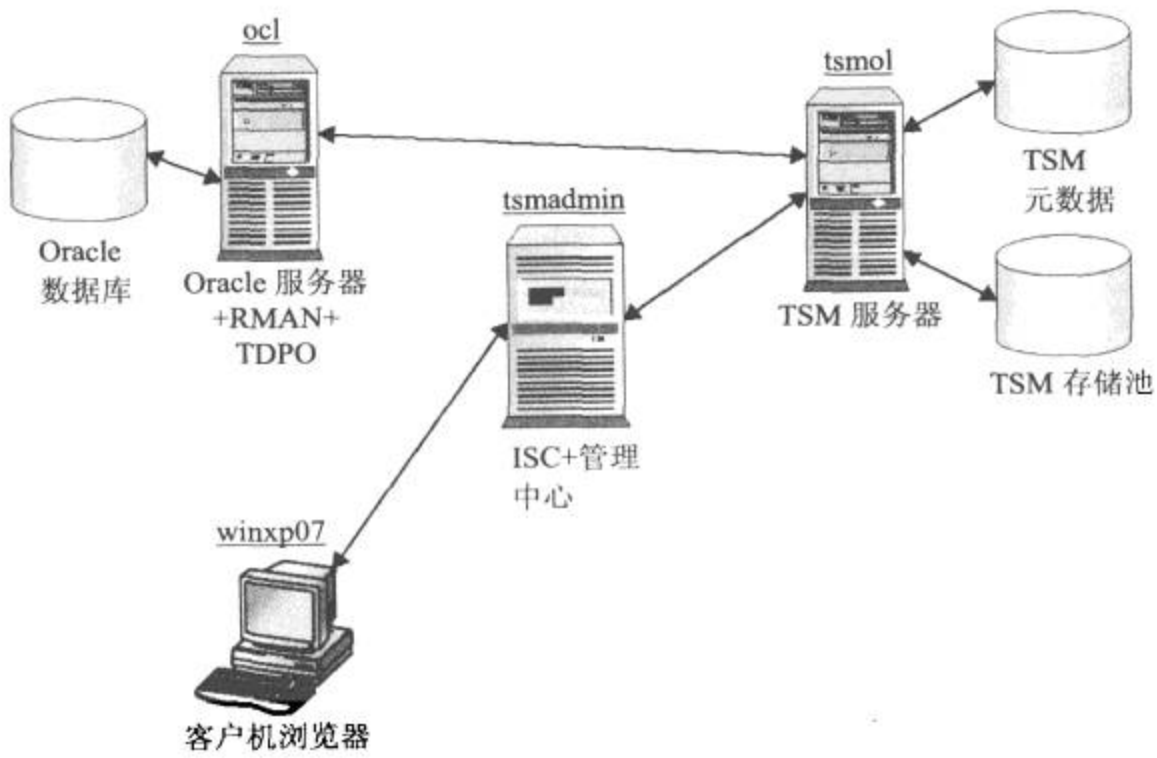


图 9-1　TSM 体系结构

表 9-1 概述了图 9-1 中显示的节点。在本章的示例中将使用这些节点来显示如何跨网络分布 TSM 组件。

表 9-1　TSM 节点名和角色

节　点　名	操 作 系 统	角　　色
tsm01	Linux	TSM 服务器
tsmadmin	Linux	集成解决方案控制台，管理中心服务器
ocl	Linux	Oracle 数据库，Tivoli Data Protection for Oracle；TSM 客户端
winxp07	Windows XP	集成的解决方案控制台/管理中心 Web 客户端

表 9-2 列出并简要描述了用于服务器 tsm01 上的磁盘设备，这些磁盘设备用于 Oracle RMAN 备份。

表 9-2　TSM 存储设备的裸磁盘

物理设备名	Linux 加载点	容　　量	目　　标
/dev/sda1	/tsm01	3.5GB	Oracle 备份池的磁盘 1
/dev/sdb1	/tsm02	3.5GB	Oracle 备份池的磁盘 2
/dev/sdc1	/tsm03	3.5GB	Oracle 备份池的磁盘 3
/dev/sdd1	/tsm04	3.5GB	Oracle 备份池的磁盘 4

9.1.1　TSM 服务器系统对象

利用 TSM 服务器中系统对象的多级结构，可以以最优的方式方便地配置环境中各种数据源的备份。基于相同的原因，利用这种灵活性的层次结构也可以很容易地将特定的配置分配给无关的数据源。图 9-2 显示了 TSM 系统对象之间的关系，以及任意给定 TSM 服务器上客户端所使用的对象其类型和数量之间的关系。

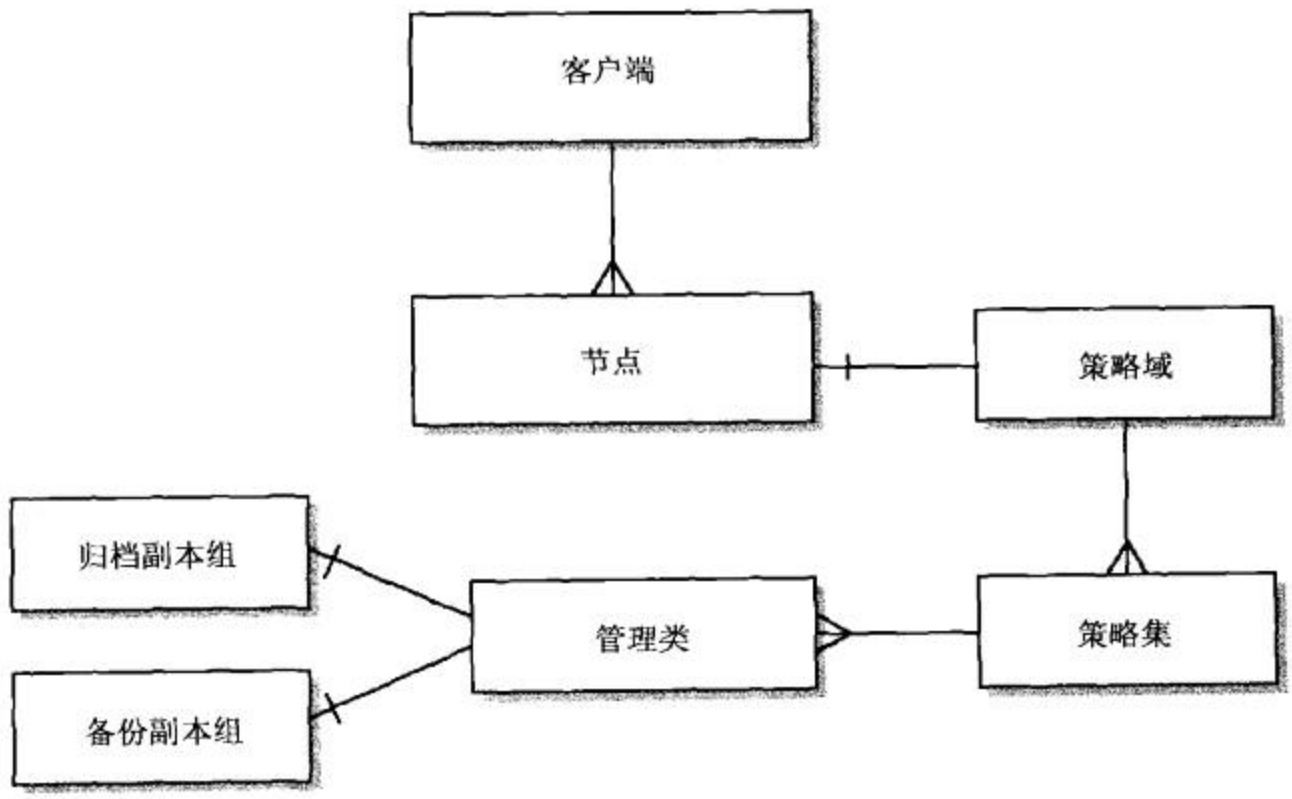


图 9-2　客户端/TSM 关系和 TSM 系统对象

在图 9-2 中，最高层次是策略域：策略域由一个或多个策略集组成，每个策略集则由一个或多个管理类组成。每个管理类可以有一个归档副本组和一个备份副本组。下面的章节中将详细介绍每个对象。

1.　策略域

策略域是具有相似备份和归档数据需求的一组客户端。可以将策略域用于特定部门中的每个人、或者特定建筑或楼层中的每个人、或者特定文件服务器的所有用户。

默认的 TSM 安装包括一个称为 standard 的默认策略域。在本章后面的示例中，将使

用 standard 策略域。将备份客户端分配到策略域。

2. 策略集

策略集是一组管理类。每个策略域包含一个或多个策略集，但在任何时候，策略域中只有一个策略集处于活动状态。使用策略集可以在可用管理类之间方便地切换。

3. 管理类

管理类是零个、一个或两个副本组的集合。可以将策略集中的一个管理类指定为默认管理类。一般使用管理类划分客户端数据，划分主要根据数据对业务的关键程度、更改的频繁程度、或者是否必须长期地保留来确定。管理类最多可以有一个备份副本组和一个归档副本组。

4. 备份副本组和归档副本组

副本组指定了控制备份或归档文件以下特征的常见属性。

- Generation：用于指定要保存的文件副本个数。
- Destination：用于指定备份将包含在哪个存储池中。
- Expiration：用于指定文件何时因为超期或过了保存周期而被删除。

备份副本组包含的属性对以下方面进行控制：从最近一次备份以来文件是否已经改动、再次备份文件所经过的天数、如何处理用于备份的文件。相对地，归档副本组则包含控制如下方面的属性：是否归档使用中的文件、服务器在何处存储文件的归档副本，以及服务器保存文件归档副本多长时间。TDPO 只将备份副本组用于 Oracle 备份。

9.1.2 TSM 客户端

TSM 的客户端部分(包括 TSM API)，需要在使用 TSM 服务器进行备份或恢复的服务器上安装。Oracle 服务器上的软件安装也包括 TSM 的 RMAN 库接口：Tivoli Data Protection for Oracle(TDPO)。

实用例

假定您为一个网站开发及托管服务公司工作。该公司的一些 Oracle 数据库使用采用公共管理类的 TDPO 和 TSM 执行备份作为生产数据库，备份存储在磁带上，保存周期为 30 天。备份建立的第二天将会被复制到异地存储位置。同时，开发商也使用一些非生产数据库用于开发网站。

在星期一上班后，发现非生产数据库上的硬件出现故障。磁盘故障导致一些联机重做日志和所有控制文件丢失。由于没有对非生产数据库进行备份，所以花费几天的时间才完成数据库重建工作。而开发商花费了更多的时间重做数据库变化情况。在此之后，管理层开始考虑对开发数据库进行备份。为了减少磁带备份的开销，备份周期改为 7 天，而且将不采用异地存储。

通过仔细研究，确定满足上述需求且不影响生产数据库备份的最佳方法是建立第二

个管理类。要求 TSM 管理员建立一个新的有 7 天保存期且不需要异地复制的管理类。在备份非生产数据库时，您引用了该管理类。非生产数据库可以被备份并保存 7 天，不执行异地存储，而生产数据库不会受到影响。

使用 TDPO，RMAN 可以将如下数据库对象备份到 TSM：

- 数据库
- 表空间
- 数据文件
- 归档日志文件
- 控制文件
- Pfiles 和 SPfiles 文件

此外，可以在数据库脱机时执行完全数据库还原，也可以在数据库联机或脱机时执行表空间或数据文件还原。

服务器 oc1 是图 9-1 中的 Oracle 实时应用集群(Real Application Cluster，RAC)数据库中的客户端节点，并且是服务器 tsm01 上 TSM 的客户端。

9.1.3　TSM Administration Center 和 Web 客户端

Administration Center(管理中心)是基于 Web 的界面，使用该界面可实现对 IBM TSM 版本 5.3 服务器的集中配置和管理。由于它被当作 IBM 集成解决方案控制台(Integrated Solutions Console，简写为 ISC)组件来安装——因此，可以使用 ISC 以一种常见的管理界面来管理大量不同类型的系统和应用程序。

在图 9-1 中，服务器 tsmadmin 驻留 ISC 和 Administration Center 插件。TSM 管理员使用工作站 winxp07 上的 Web 浏览器来连接 tsmadmin 上的 ISC，ISC 则发送控制台命令并从 TSM 服务器 tsm01 接收状态信息。可以使用该方法从具有与服务器 tsmadmin 存在网络连接的任何 Web 浏览器中管理 TSM。

RMAN 实习：为 Oracle 配置 TDPO

实习注意事项

为实习下列工作，需要由一个可操作的 TSM 服务器以及客户端环境，并且安装了 Oracle 数据库。

步骤 1　为在 Oracle 服务器上安装 TDPO，需要安装下列 RPM 包：

- TIVguid.i386.rpm：为该服务器建立一个全局唯一的标识符(GUID)，以区分其他所有访问 TSM 的服务器。
- TIVsm-API.i386.rpm：安装应用程序接口(API)库以支持 TPDO 或其他应用程序能够通过编程访问 TSM。
- TDP-Oracle.i386.rpm：包含 Oracle TMAN 用于链接 TSM 的库及链接定义。

以下是安装 TDP-Oracle 后显示的结果。

```
[root@oc1 DPO]# rpm -i TDP-Oracle.i386.rpm
Post Installation of IBM Tivoli Storage Manager for Databases - Oracle.
Checking Tivoli Signature File.
Created Tivoli Signature File.
Creating symbolic links
created link /opt/tivoli/tsm/client/oracle/bin/libobk.so
      /usr/lib/libobk.so
created link /opt/tivoli/tsm/client/oracle/bin/tdpoconf /usr/bin/tdpoconf

Post Installation of IBM Tivoli Storage Manager
for Databases - Oracle Complete.
Be sure to set up the system configuration file
before starting the client!
[root@oc1 DPO]#
```

步骤 2　将客户端 oc1 注册到 TMS 服务器上。

```
TSM:SERVER1> reg node oc1_oracle orabakpw maxnummp=2 passexp=0
ANR2017I Administrator SERVER_CONSOLE issued command:
     REGISTER NODE oc1_oracle ?***? maxnummp=2 passexp=0
ANR2060I Node OC1_ORACLE registered in policy domain STANDARD.
ANR2099I Administrative userid OC1_ORACLE defined
     for OWNER access to node OC1.
TSM:SERVER1>
```

注意设置了 maxnummp=2:该设置表明客户端将数据备份到磁带时可使用的并行会话的最大数量。即使我们在这些例子中采用磁盘驱动器作为备份目标，定义并行数量以适应采用磁带备份的情况发生仍然是一个值得采用的好想法。

注册客户端节点同时也就建立了一个可用于链接到 TSM 服务器的管理账户。然而，为每个管理员建立一个独立的服务器账户，能够使您更好地控制分派给每个管理员的权限，同时也可在管理员改变 TSM 服务器配置时，得到更多的审计信息。

下一步骤是建立服务器 oc1 上的 Oracle 数据库到 tsm01 上的 TSM 服务器之间的链接。通过下列步骤实现该链接：

(1) 在 tdpo.opt 配置文件中定义 TDPO 选项。
(2) 建立 dsm.opt。
(3) 建立 dsm.sys。
(4) 在 TSM 服务器上注册 TDPO 节点并定义其他策略需求。
(5) 配置 TSM 复制组选项。
(6) 为 TSM 服务器建立口令文件。
(7) 在 Oracle 库目录上建立符号链接。
(8) 测试 TDPO 连接。

步骤 3　定义 TDPO 选项。在 Oracle 客户端节点 oc1 上，进入/opt/tivoli/tsm/client/oracle/bin 目录，复制 tdpo.opt.smp(示例文件)到 tdpo.opt。tdpo.opt 文件，您可能会猜到，定义了与 TDPO 有关的选项，例如 TDPO 如何连接到 TSM 服务器。在下面的代码中，

取消以 TDPO_NODE 开始的注释行，用 TSM 客户端节点名替换<hostname>，在示例中，采用的 TSM 客户端节点是 oc1_oracle。另外，取消以 DSMI_ORC_CONFIG 和 DSMI_LOG 开始的注释行，如果安装 TDPO 的目录不同于默认目录。tdpo.opt 文件如下所示：

```
************************************************************
* IBM Tivoli Storage Manager for Databases
*   Data Protection for Oracle
*
* Sample tdpo.opt for the Linux86 Data Protection for Oracle
************************************************************
*DSMI_ORC_CONFIG      /opt/tivoli/tsm/client/oracle/bin/dsm.opt
*DSMI_LOG             /opt/tivoli/tsm/client/oracle/bin

*TDPO_FS              /adsmorc
TDPO_NODE             oc1_oracle
*TDPO_OWNER           <username>
*TDPO_PSWDPATH        /opt/tivoli/tsm/client/oracle/bin

*TDPO_DATE_FMT        1
*TDPO_NUM_FMT         1
*TDPO_TIME_FMT        1

*TDPO_MGMT_CLASS_2    mgmtclass2
*TDPO_MGMT_CLASS_3    mgmtclass3
*TDPO_MGMT_CLASS_4    mgmtclass4
```

步骤 4　建立 dsm.sys。文件 dsm.sys 定义了连接到每个 TSM 服务器的方式，定义了端口号，TCP/IP 地址等等。将文件/opt/tivoli/tsm/client/api/bin/dsm.sys.smp 复制到/opt/tivoli/tsm/client/oracle/bin/dsm.sys 并按照如下示例修改相关值。

```
************************************************************************
* Tivoli Storage Manager *
* *
* Sample Client System Options file for UNIX (dsm.sys.smp) *
************************************************************************

* This file contains the minimum options required to get started
* using TSM. Copy dsm.sys.smp to dsm.sys. In the dsm.sys file,
* enter the appropriate values for each option listed below and
* remove the leading asterisk (*) for each one.

* If your client node communicates with multiple TSM servers, be
* sure to add a stanza, beginning with the SERVERNAME option, for
* each additional server.
************************************************************************
SErvername            SERVER1
COMMmethod            TCPip
TCPPort               1500
```

```
TCPServeraddress    192.168.2.69
Passwordaccess      prompt
```

IP 地址 192.168.2.69 是服务器 tsm01 的地址。为避免手动为每个备份输入口令，可以使用本章后面将要介绍的 tdpoconf 实用程序建立一个口令文件，TDPO 将使用该文件与 TSM 服务器进行认证。

步骤 5 建立 dsm.opt。文件 dsm.opt 定义了在该节点上备份时使用的 TSM 服务器名。在目录/opt/tivoli/tsm/client/oracle/bin 中，创建只包含一行内容的文件，如下所示：

```
SERVERNAME SERVER1
```

步骤 6 注册到 TDPO 节点。在 TSM 控制台上使用如下命令向 TSM 注册 Oracle 服务器：

```
reg node oc1_oracle orabakpw maxnummp=2 passexp=0
```

使用如下控制台命令更新 TSM 节点上的 oc1_oracle 节点的配置，启用 RMAN 目录的归档和到期设置以控制 TSM 服务器上的备份保存：

```
update node oc1_oracle backdelete=yes
```

步骤 7 配置 TSM 复制组选项。因为 RMAN 为所创建的每个备份文件指定了不同的备份文件名，所以保存到 TSM 备份存储池的备份对象具有唯一的文件名，从而使它们永远不会到期。因此，我们必须设置复制组属性 verdeleted 为 0，从而在 RMAN 命令或策略设置备份对象为“不活动的”或“到期”的状态时，TDPO 可以从 TSM 备份存储池中删除这些不需要的备份对象。参数 verdeleted 的值代表当文件已经从客户端中删除时，可保存的该文件副本的最大备份数量；因此，设置该值为 0 可确保 TSM 服务器上到期的备份文件在下一次到期处理执行时被删除。

在该示例中，使用 TDPO 备份的默认复制组，因此设置选项 verdeleted 如下：

```
TSM:SERVER1> update copygroup standard standard standard verdeleted=0
ANR2017I Administrator SERVER_CONSOLE issued command:
     UPDATE COPYGROUP standard standard standard verdeleted=0
ANR1532I Backup copy group STANDARD updated in policy domain
     STANDARD, set STANDARD, management class STANDARD.
TSM:SERVER1>
```

步骤 8 建立 tdpo 口令文件。为了确保不需要交互式地为 TSM 服务器上的每个 RMAN 备份指定口令，使用如下所示的 tdpoconf 实用程序：

```
[root@oc1 bin]# tdpoconf password

IBM Tivoli Storage Manager for Databases:
Data Protection for Oracle
Version 5, Release 2, Level 0.0
(C) Copyright IBM Corporation 1997, 2003. All rights reserved.
```

```
*************************************************************
*       IBM Tivoli Storage Manager for Databases Utility    *
* Password file initialization/update program               *
*     ROOT privilege needed to update value                 *
*************************************************************

Please enter current password:
Please enter new password:
Please reenter new password for verification:

ANU0260I Password successfully changed.
[root@oc1 bin]#
```

tdpoconf 实用程序创建或更新名为/opt/tivoli/tsm/client/oracle/bin/TDPO.oc1_oracle 的加密口令文件。

步骤 9　除了在安装程序包 TDP-Oracle.i386.rpm 期间创建的链接外，需要在 Oracle 的默认库目录中创建 TSM 库函数的符号链接，如下所示：

```
ln /opt/Tivoli/tsm/client/oracle/bin/libobk.so
      $ORACLE_HOME/lib/libobk.so
```

由于 RPM 实用程序不知道 Oracle 可执行程序和库存储位于何处，因此必须手动定义这个链接。如果存在多个 Oracle 主目录，则必须为每个想使用 TDPO 的主目录建立链接。

Libobk 是库文件的通用名。每个希望与 RMAN 接口的备份提供商都会提供一个与 Libobk 链接的库文件。

步骤 10　测试 TDPO 连通性。使用 sbttest 实用程序确保可以建立 TSM 服务器的连接。可以在目录$ORACLE_HOME/bin 中找到 sbttest。为了运行 sbttest，需要设置环境变量 TDPO_OPTFILE 以指向前面创建的 tdpo.opt 配置文件，然后运行 sbttest test 命令，如同下面的示例所示：

```
[oracle@oc1 bin]$ export
     TDPO_OPTFILE=/opt/tivoli/tsm/client/oracle/bin/tdpo.opt
[oracle@oc1 bin]$ sbttest test
The sbt function pointers are loaded from libobk.so library.
-- sbtinit succeeded
Return code -1 from sbtinit, bsercoer = 0, bsercerrno = 0
Message 0 not found; product=RDBMS; facility=SBT
[oracle@oc1 bin]$
```

Output-Sbtinit-Succeeded 和命令返回的代码-1 都表明该测试已成功运行。

9.2　使用 TDPO 完成 RMAN 备份

完成了 TDPO 设置后，就为执行 RMAN 备份做好了准备。我们将在 RMAN 会话中

使用 allocate channel 命令来定义备份位置；虽然信道类型总为 sbt_tape，但在 TSM 服务器上的实际备份设备也可能是磁盘、可写入的 DVD 或者物理磁带驱动器；RMAN 不知道，实际上您也不会关注备份包含哪些物理设备。只要可以从灾难中恢复系统，采用哪种物理设备来备份并不十分重要。

在第一个示例中，只将 USERS 表空间备份到 TSM：

```
[oracle@oc1 ~]$ rman target /

Recovery Manager: Release 11.2.0.1.0 - Production on Tue Jan 26 11:17:01 2010

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

connected to target database: RAC (DBID=2170964680)

RMAN> run
2> { allocate channel tdpo type 'sbt_tape' parms
3>      'ENV=(TDPO_OPTFILE=
4>        /opt/tivoli/tsm/client/oracle/bin/tdpo.opt)';
5>  backup tablespace users;
6>  release channel tdpo;
7> }

using target database control file instead of recovery catalog
allocated channel: tdpo
channel tdpo: sid=293 instance=rac1 devtype=SBT_TAPE
channel tdpo: Tivoli Data Protection for
     Oracle: version 5.5.1.0

Starting backup at 22-JUL-06
channel tdpo: starting full datafile backupset
channel tdpo: specifying datafile(s) in backupset
input datafile fno=00004
     name=+DATA/rac/datafile/users.259.582982545
channel tdpo: starting piece 1 at 26-JAN-2010
channel tdpo: finished piece 1 at 26-JAN-2010
piece handle=02horjvc_1_1 tag=TAG20060722T212604
     comment=API Version 2.0,MMS Version 5.2.0.0
channel tdpo: backup set complete, elapsed time: 00:00:03
Finished backup at 25-JAN-2010

released channel: tdpo
RMAN>
```

在备份到 TSM 时，唯一需要额外做的工作是在 RMAN 的 env 参数中指定 TDPO 选项文件的位置。在下面第二个示例中，我们备份整个数据库：

```
RMAN> run
2> { allocate channel tdpo type 'sbt_tape' parms
```

```
3>     'ENV=(TDPO_OPTFILE=
4>           /opt/tivoli/tsm/client/oracle/bin/tdpo.opt)';
5>   backup database;
6>   release channel tdpo;
7> }

allocated channel: tdpo
channel tdpo: sid=293 instance=rac1 devtype=SBT_TAPE
channel tdpo: Tivoli Data Protection for Oracle: version 5.5.1.0

Starting backup at 26-JAN-2010
channel tdpo: starting full datafile backupset
channel tdpo: specifying datafile(s) in backupset
input datafile fno=00003 name=+DATA/rac/datafile/sysaux.257.582982545
input datafile fno=00001 name=+DATA/rac/datafile/system.256.582982545
input datafile fno=00002 name=+DATA/rac/datafile/undotbs1.258.582982545
input datafile fno=00005 name=+DATA/rac/datafile/example.264.582982703
input datafile fno=00006 name=+DATA/rac/datafile/undotbs2.265.582982943
input datafile fno=00007 name=+DATA/rac/datafile/undotbs3.266.582983003
input datafile fno=00004 name=+DATA/rac/datafile/users.259.582982545
channel tdpo: starting piece 1 at 26-JAN-2010
channel tdpo: finished piece 1 at 26-JAN-2010
piece handle=03hork9s_1_1 tag=TAG20060722T213140 comment=API
    Version 2.0,MMS Version 5.5.1.0
channel tdpo: backup set complete, elapsed time: 00:03:26
channel tdpo: starting full datafile backupset
channel tdpo: specifying datafile(s) in backupset
including current control file in backupset
including current SPFILE in backupset
channel tdpo: starting piece 1 at 26-JAN-2010
channel tdpo: finished piece 1 at 26-JAN-2010
piece handle=04horkga_1_1 tag=TAG20060722T213140 comment=API
    Version 2.0,MMS Version 5.5.1.0
channel tdpo: backup set complete, elapsed time: 00:00:06
Finished backup at 26-JAN-2010

released channel: tdpo

RMAN>
```

需要注意的是，不需要指定备份的目标位置或使用的磁盘设备。TSM 自动将备份文件放入一个或多个存储池卷。

```
RMAN> list backup;

using target database control file instead of recovery catalog
List of Backup Sets
===================
```

```
BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
1       Full    2.00M      SBT_TAPE    00:00:02     26-JAN-2010
        BP Key: 1 Status: AVAILABLE    Compressed: NO Tag:
                                                    TAG20060722T212604
        Handle: 02horjvc_1_1 Media:
List of Datafiles in backup set 1
File LV Type Ckp SCN Ckp Time Name
---- -- ---- ---------- --------- ----
4       Full  8772169    26-JAN-2010
                         +DATA/rac/datafile/users.259.582982545

BS Key Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
2       Full    1.24G      SBT_TAPE    00:03:24     26-JAN-2010
        BP Key: 2   Status: AVAILABLE Compressed: NO Tag:
                                                   TAG20060722T213140
       Handle: 03hork9s_1_1 Media:
List of Datafiles in backup set 2
File LV Type Ckp SCN Ckp Time Name
---- -- ---- ---------- --------- ----
1       Full 8772449    26-JAN-2010
                        +DATA/rac/datafile/system.256.582982545
2       Full 8772449    26-JAN-2010
                        +DATA/rac/datafile/undotbs1.258.582982545
3       Full 8772449    26-JAN-2010
                        +DATA/rac/datafile/sysaux.257.582982545
4       Full 8772449    26-JAN-2010
                        +DATA/rac/datafile/users.259.582982545
5       Full 8772449    26-JAN-2010
                        +DATA/rac/datafile/example.264.582982703
6       Full 8772449    26-JAN-2010
                        +DATA/rac/datafile/undotbs2.265.582982943
7       Full 8772449    26-JAN-2010
                        +DATA/rac/datafile/undotbs3.266.582983003

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
3       Full    14.75M     SBT_TAPE    00:00:05     26-JAN-2010
        BP Key: 3   Status: AVAILABLE  Compressed: NO Tag:
                                                   TAG20060722T213140
        Handle: 04horkga_1_1 Media:
  Control File Included: Ckp SCN: 8772600 Ckp time: 26-JAN-2010
  SPFILE Included: Modification time: 26-JAN-2010

RMAN>
```

最后，通过查看存储池卷的属性，可以看到备份使用了存储池中的多少磁盘空间，如图 9-3 所示。单击图 9-4 中的 ORACLEPOOL 存储池链接，然后单击图 9-5 中的 Volumes 链接，即可访问该页面。

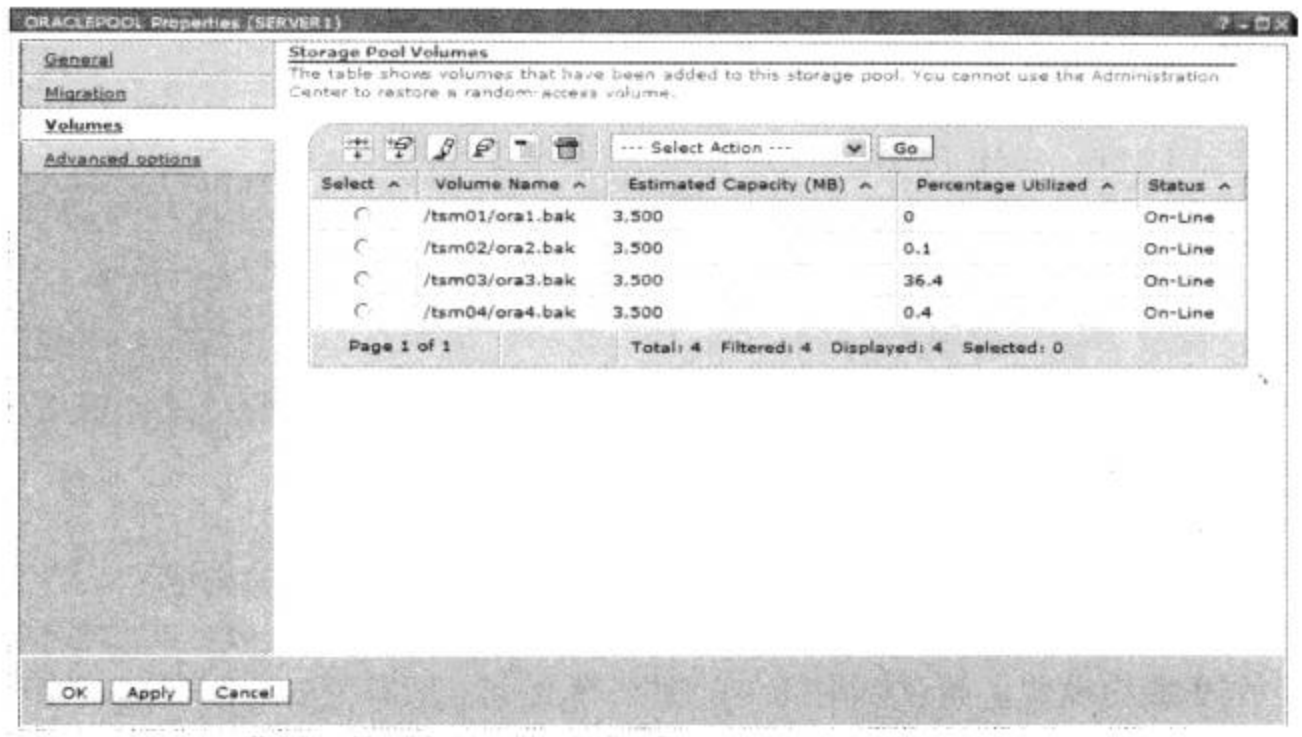


图 9-3　查询存储池卷

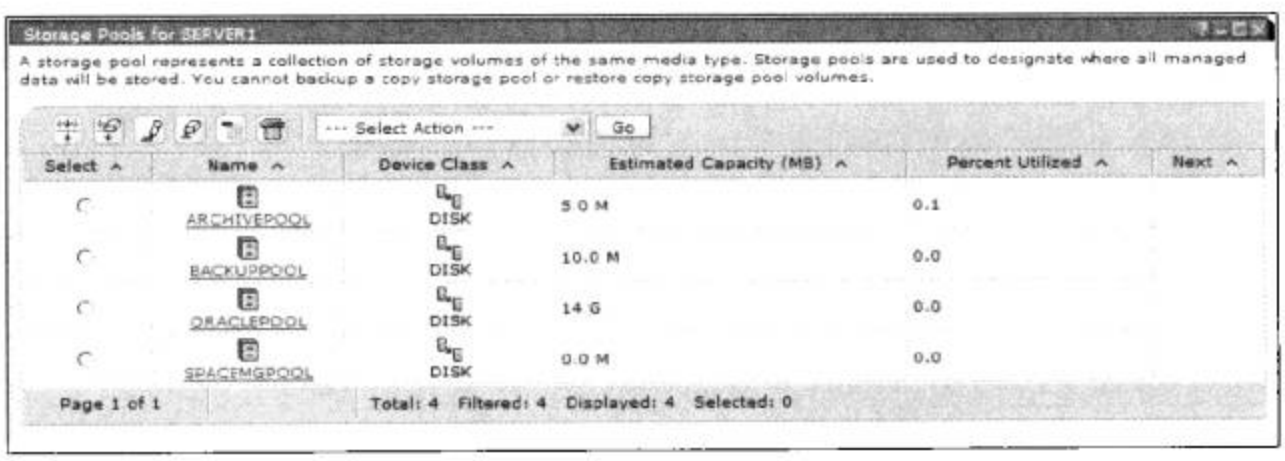


图 9-4　显示存储池及容量

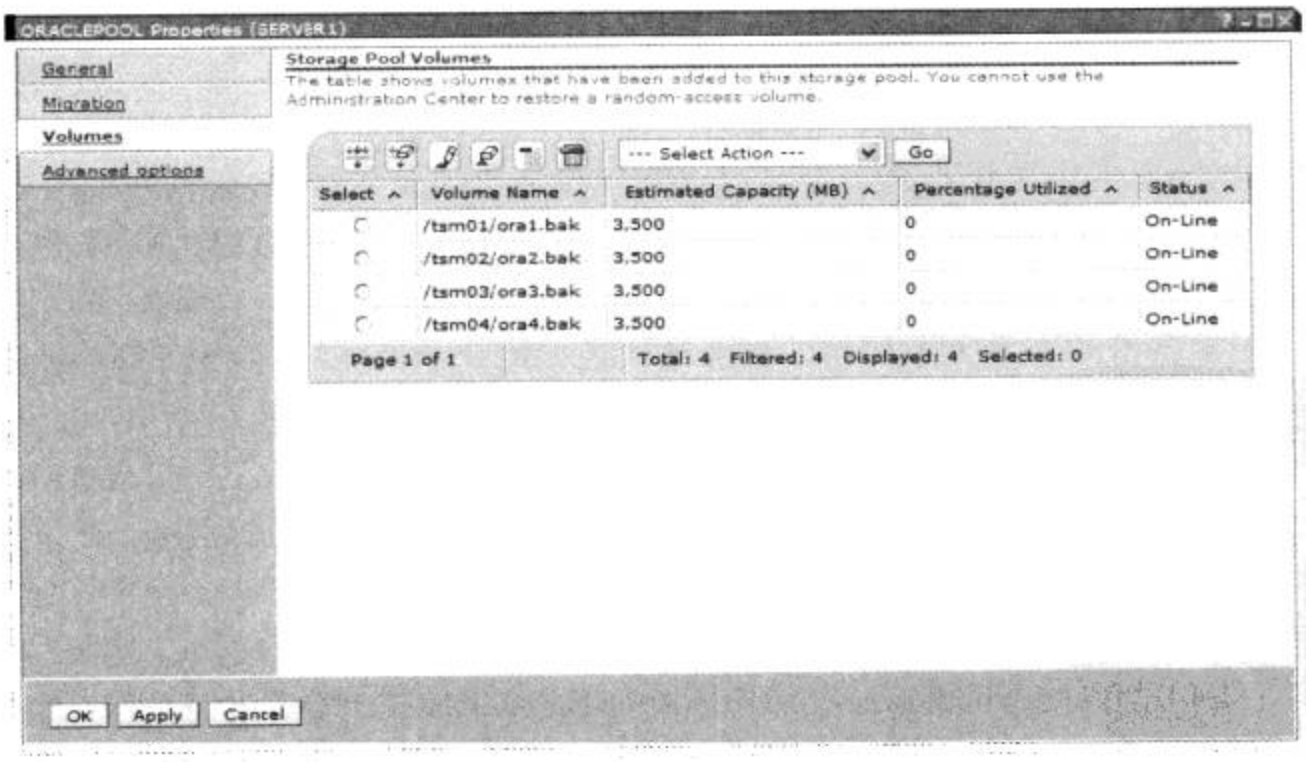


图 9-5　客户端显示存储池卷

名称代表什么

TDPO_NODE 可以称作任何事物。在许多 Oracle 数据库环境中，都使用 TSM 和 TDPO 来实现备份，TDPO_AME 可能是不能进行备份和还原的根本原因。如果使用 oc1_oracle 的 TDPO_NODE 进行备份，则会创建一个新的 TDPO_NODE，oc2_oracle；如果修改 tdpo 文件以使用新的 TDPO_NAME，那么在将 tdpo 文件中的 TDPO_NODE 改为 oc1_oracle 以前，RMAN 就不能访问 oc1_oracle 下的任何备份。另一个常见的问题是所使用的 TDPO_NODE 的口令文件并没有在数据库服务器上。这将导致备份与还原操作失败。

默认信道

由于每个数据库管理员都知道如何手动分配信道，因此可能会造成一些容易发生的输入错误。如果系统中包含多个数据库，可能造成不知道使用哪个 tdpo 文件的情况发生。通过对一个参数的设置，就可以用简单的数据库命令，无需手动分配信道，实现对数据库的备份工作。在 RMAN>提示符中运行 show all 命令可以显示各种已配置的 RMAN 选项。为实现备份到 TSM 使用自动信道分配方法，只需要考虑两个选项：CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE'和 CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS：

通过 RMAN 提示符运行下列命令，就可以设置默认信道。

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE';
RMAN> CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
ENV=(TDPO_OPTFILE=/opt/tivoli/tsm/client/oracle/bin/tdpo.opt)' FORMAT %U';
```

运行配置命令之后，备份就显得更加容易了：

```
RMAN> backup database;
```

9.3　删除数据库备份

使用 RMAN 的主要优点之一是其提供的删除备份的公共接口。无论是使用磁带备份还是磁盘备份，都可以使用同样的命令删除过期的备份。在使用 TSM 和 TDPO 时，即使删除备份的命令相同，仍然需要花些时间完成额外的清理工作。在删除备份时，将会注意到，无论备份的容量大小如何，都会被快速删除。当 RMAN 要求 TSM 删除备份文件时，TSM 并不会立即删除该文件，因为此时磁带可能不可用，也可能所有的磁带驱动器正非常忙碌。所以 TSM 将需要删除的文件标记为将被删除状态，并向 RMAN 返回一个删除成功的代码。TSM 管理员需要执行一个净化(purge)进程以实际删除该文件。如果使用了一个由 TSM 提供的包含 tdposync 工具的目录，就可以比较 RMAN 目录显示的已删除的文件和 TSM 已经删除的文件的情况。从默认位置执行 tdposync，使用示例 tdpo 文件，运行/opt/tivoli/tsm/client/oracle/bin/tdposync SYNCDB–TDPOfile=/opt/tivoli/tsm/ client/oracle/bin/tdpo.opt。必须包含用于与备份比较的 tdpo 文件。执行该操作后，tdposync 将

会要求输入用户名，口令，以及到 RMAN 目录的连接字符串。若有多个 RMAN 目录包含备份记录，则添加一个参数-NUMCATALOG=number。若比较后发现存在差异，tdposync 将显示列表并在从 TSM 删除文件前要求确认。

9.4　公共备份场景的故障诊断

使用 TSM 备份 Oracle 数据库包含四个部分：RMAN，Oracle，O/S 和 TSM—每个部分都可能出现故障并报告错误。当首先使用 RMAN 控制台时，要知道哪个部分正在报告错误是非常困难的事情，因为所有的错误都返回到 RMAN 控制台上。在下面的示例中，显示的是在执行一个简单的备份数据库任务后所返回的 O/S，Oracle，以及 RMAN 错误报告。

```
RMAN> backup database;
    RMAN-00571: ===========================================================
    RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
    RMAN-00571: ===========================================================
    RMAN-03002: failure of backup command at 07/17/2009 10:34:31
    RMAN-03014: implicit resync of recovery catalog failed
    RMAN-06403: could not obtain a fully authorized session
    ORA-01034: ORACLE not available
    ORA-27101: shared memory realm does not exist
    Linux-x86_64 Error: 2: No such file or directory
    RMAN>
```

在阅读错误消息时最好使用自下向上的方法。

6. RMAN-03002：备份命令的故障；

5. RMAN-03014：恢复目录隐含的再同步的故障；

4. RMAN-06403：不能获得一个完整授权的会话；

3.ORA-01034：Oracle 不可用；

2, ORA-27101：共享内存区不可用；

1. Linux-x86_64 Error:2：无该文件或目录。

第 1 行是一个返回到 Oracle 的 o/s 错误。在此情况下，解决该问题非常不乐观。第 2，3 行对 Oracle DBA 提供了答案，DBA 应该理解。“ORA-27101：共享内存区不可用”意味着数据库实例未运行，因此 RMAN 无法连接到数据库。

无论是否使用 TSM，第 1 个示例都可能发生。下一个场景将介绍一个仅会在使用 TSM 时出现的错误。首先，RMAN 运行分配信道的数据块，然后备份归档日志，删除它们，最后释放信道。

```
RMAN> RMAN> run
    2> { allocate channel tdpo type 'sbt_tape' parms
    3>     'ENV=(TDPO_OPTFILE=
    4>           /opt/tivoli/tsm/client/oracle/bin/tdpo.opt)';
    5>   backup database;
```

```
6>   release channel tdpo;
7> }

RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
ORA-19506: failed to create sequential file, name="rac_9ikk7ac3_1_1", parms=""
ORA-27028: skgfqcre: sbtbackup returned error
ORA-19511: Error received from media manager layer, error text:
ANS1353E (RC53) Session rejected: Unknown or incorrect ID entered

RMAN>
```

仍然采用自下向上的方法阅读，我们首先阅读第 1 行错误消息。

4. ORA-19506：建立连续文件错误，name="a400_9ikk7ac3_1_1", parms=" "

3. ORA-27028：skgfqcre: sbtbackup 返回错误

2. ORA-19511：来自介质管理层的错误，错误文本：

1. ANS1353E(RC53) Session rejected：输入的 ID 未知或不正确

ANS 或 AUN 是从 TDPO 库返回的错误。第 3 行指示 RMAN 从介质管理层(TDPO 库)接收到一个错误。第 2 行和第 1 行仍然来自 RMAN，指示 RMAN 不能为归档日志建立备份文件。ANU 和 ANS 错误需要花大量时间进行故障检测，因为需要检查大多数配置以及参数文件以及设置，并且有大量不同的 TSM 错误返回。分析第 1 步返回的“Session rejected”错误消息，可以知道存在一个试图从 RMAN 建立到 TSM 的连接，该会话被 TSM 拒绝。导致会话被拒绝的原因很多，在上例环境中，是因为没有为 tdpo 节点创建的口令文件。正如在本章前面介绍的那样，要备份或还原 RMAN 备份，需要向 TSM 提供 tdpo 节点名和口令。为快速诊断该错误，则要检查 tdpo 选项文件中 RMAN 正在使用的 tdpo 文件名，然后检查确保 RMAN 能够访问该口令文件以获取 tdpo 名。同时在 dsm.sys 中查找参数 ERRORLOGNAME。该参数将指示文件的路径，TDPO 将在此保存错误信息且可能会提供更多的细节。另外，存储管理员在 TSM 日志错误中也可能会看到会话是否被拒绝的消息，该消息可能会提供其他的细节。

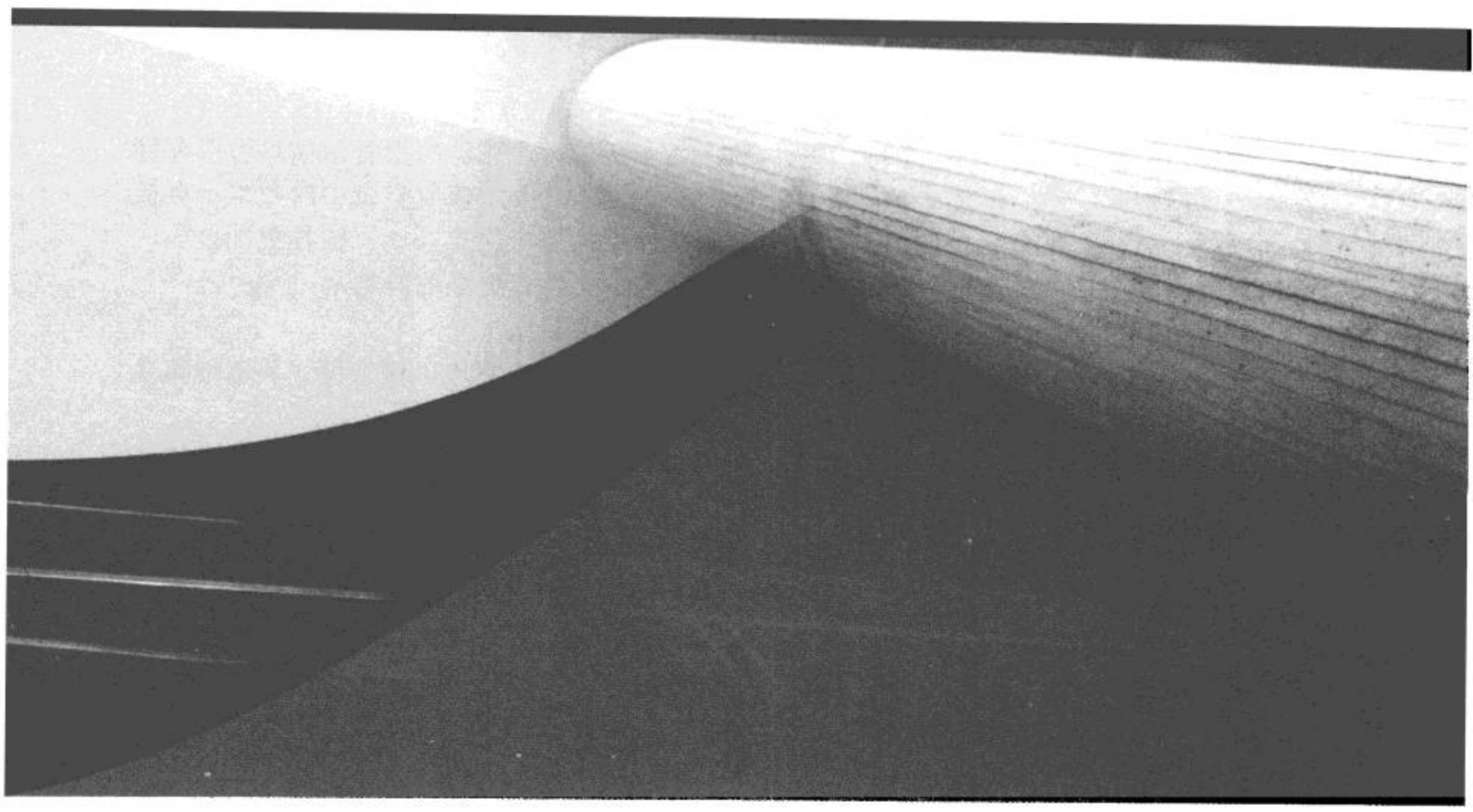
9.5　其他故障诊断

打开或关闭调试查询可简单地使用 debug on；命令或 debug off；命令。一旦打开调试程序，则需要通知 RMAN 将收集何种其他的调试信息。在运行 allocate channel 命令时，trace=1 用于在 user_dump_destination 目录中创建跟踪文件，debug=2 将指示 RMAN 将其他信息发送到 sbtio.log 文件中。

9.6　小结

一旦通过一些很容易的步骤执行了 TSM 和 TDPO 的初始安装，就可以“设置它并忘记它”，从而使 DBA 可以更多地关注于 RMAN 脚本自身，而不是关注 TSM 将备份存储在何处以及如何存储等。当发生故障时，适当地读取错误消息，并快速诊断备份故障，判断其故障是源于数据库还是 TSM。

TSM 和 TDPO 不仅能让用户很容易地使用熟悉的 RMAN 界面来备份 Oracle 数据库，还减少了企业的存储管理成本，因为可以使用一个存储管理器(Tivoli Storage Manager)来满足所有备份、恢复和归档需求。



第 10 章

恢复目录的使用

Oracle 在 RMAN 存储库(repository)中维护与 RMAN 操作相关联的所有元数据。RMAN 存储库总是存储在目标数据库的控制文件中。在某些情况下，可能希望将目标数据库的 RMAN 存储库存储在其他位置上。该位置称为 RMAN 恢复目录。

RMAN 的大多数操作并不需要恢复目录，因此，恢复目录实际上是一个可选组件。由于恢复目录通常是可选的组件，因此 RMAN 默认配置中没有恢复目录。在本章中，首先将讨论恢复目录的含义以及在何种情况下需要恢复目录。然后，讨论如何建立恢复目录以及恢复目录的备份与恢复问题。

10.1 恢复目录的含义

恢复目录是 RMAN 的一个可选组件。RMAN 备份时产生的历史备份信息可以存储在恢复目录中。与放在数据库控制文件中的 RMAN 信息不同，恢复目录中的数据不会被定期清除。因此，恢复目录的历史信息与控制文件中的历史信息相比，往往更加全面，可追溯的时间更长。使用恢复目录比使用数据库控制文件有更多的好处：

- 如果要使用 RMAN 存储脚本，就必须使用恢复目录。
- 恢复目录提供存储企业 RMAN 信息的单一存储库，它可以更方便、灵活地将企业的备份信息集中存储。
- 使用恢复目录除了可以提供当前目标数据库的报告外，还可以提供目标数据库在其他时间的报告，恢复目录使得数据库报告更具灵活性。
- 使用恢复目录，可以永久保留某些默认的数据库 RMAN 信道配置信息。

如果非常熟悉 RMAN，可能会注意到一些常用优点没有了。首先，自版本 10g 以来，在没有恢复目录的情况下，通过执行 resetlogs 命令，Oracle 数据库非常容易实现恢复。如果使用控制文件自动备份(强烈建议)方式，就没有必要使用恢复目录为控制文件恢复了。

注意：
如果不打算使用恢复目录，则保存数据库的 DBID 的记录。虽然这不是必需的内容，但数据库的 DBID 很有用，而且使用数据库的 DBID 可以简化恢复操作。

是否应该使用恢复目录？如果只有很少的一些数据库，就不值得在恢复目录上花费额外的时间和精力。如果处于多数据库环境中，则应该考虑使用恢复目录。一般来说，使用恢复目录带来的灵活性，以及统一集中的企业报告所带来的价值，超过了使用它需要增加的维护和管理成本。使用恢复目录的缺点之一在于如果目录数据库失效，则会损失所有备份。除非已经编码备份脚本，在未使用恢复目录前保留了一个备份以防使用恢复目录时的首个备份失效。

另外，恢复目录是 Data Guard 备份环境和分割镜像(split mirror)备份的基本组成部分。在这些配置中，当备份来自备份主机的数据库时，可以认为恢复目录是最近的信息，因此它是该策略的核心内容，若维护不当则成为单点损坏。底线是需要根据环境确定是否需要恢复目录。

连接 RMAN 时，必须使用 catalog 命令行参数来指定将要建立 RMAN 与恢复目录的连接。在默认情况下，RMAN 使用 nocatalog 选项，该选项指定将不会使用恢复目录。使用 catalog 参数后，需要指定包含恢复目录对象的恢复目录模式的用户 ID 和口令。下面是一个使用 RMAN 命令行连接恢复目录的示例。

```
RMAN target='sys/robert as sysdba@robt'
    catalog='cataloguser/password@bcatalog'
```

10.1.1　创建恢复目录

如同预想的那样，在连接恢复目录之前，需要执行一些设置。首先，需要创建恢复目录用户并授予该用户适当的权限；然后，还要与恢复目录连接并创建恢复目录模式对象。下面将介绍具体步骤。

1. 配置恢复目录的数据库

恢复目录数据库应该尽可能驻留在自己单独的数据库上。然而，根据经验，许多站点使用活动数据库作为恢复目录数据库，只要在备份数据库时采取预防措施，这也是可行的。

对于将维护一年内的恢复目录记录的恢复目录数据库，Oracle 提出了如表 10-1 所示的空间分配建议。

表 10-1　空间分配建议

表 空 间 名	大 小 需 求
SYSTEM	90MB
TEMP	5MB
UNDO	5MB
RECOVERY CATALOG SCHEMA	每个注册的数据库 15MB
ONLINE REDO LOGS	每个联机重做日志文件 1MB

2. 创建恢复目录用户

通常，恢复目录应当驻留在自己单独的数据库中，这是因为如果恢复目录正好位于试图恢复的那个数据库中，恢复目录就没有多少用途了。下面的 RMAN 实习提供了一组创建恢复目录用户账户的详细指令。

RMAN 实习：创建恢复目录用户账户

实习注意事项

为了完成该 RMAN 实习，需要安装 Oracle 软件。并标识创建该恢复目录模式的数据库；此外，还需要具有对这个数据库的管理权限以便创建恢复目录用户账户。最后，要为这个恢复目录用户账户指定名称和口令。

应当为这个恢复目录模式对象创建一个表空间，这个表空间的起始大小建议设置为 15MB。

步骤 1　创建恢复目录用户。确保不要使用 SYSTEM 表空间作为临时表空间(检查新的 Oracle Database 11g 默认临时表空间特性)。将创建的恢复目录表空间(“实习注意事

项”中建议的)分配给该模式作为默认的表空间，并在这个恢复目录表空间上将恢复目录用户分配到 unlimited quota。下面是该操作的示例:

```
CREATE USER rcat_user IDENTIFIED BY rcat_password
DEFAULT TABLESPACE catalog;
```

步骤 2 向恢复目录用户授予下面的角色:

- connect
- resource
- recovery_catalog_owner

下面的示例为前面创建的 RCAT_USER 用户授予所需的角色:

```
GRANT connect, resource, recovery_catalog_owner TO rcat_user;
```

注意:
恢复目录用户账户在某种程度上也算是一个特许的数据库账户，尽可能像保护系统用户那样保护好这个账户。

3. 创建恢复目录模式对象

前面已经创建了恢复目录数据库和用户，接下来就应该创建恢复目录。创建恢复目录是 Oracle Database 11g 中的一个简单的进程，需要做的就是使用 RMAN。在启动 RMAN 时，使用 target 参数连接到目标数据库，再使用 catalog 参数连接恢复目录数据库模式(前述所创建的模式)。

然后在 RMAN 提示符下执行 create catalog 命令，此时可以选择使用 tablespace 参数来定义在其中创建 RMAN 模式对象的表空间。下面的 RMAN 实习示例使用 create catalog 命令来创建恢复目录模式。

RMAN 实习：创建恢复目录

实习注意事项

为了完成该 RMAN 实习，应当完成前一个 RMAN 实习(“创建恢复目录用户账户”)。此外，假定已经创建了一个名为 CATALOG_TBS 的表空间，将在该表空间中创建 RMAN 模式对象。

步骤 1 使用 RMAN 连接到恢复目录:

```
RMAN catalog=rcat_user/rcat_password
```

步骤 2 在 RMAN 提示符下执行如下的 create catalog 命令:

```
create catalog tablespace catalog_tbs;
```

4. 使用恢复目录注册数据库

在已经完成了使用恢复目录的准备工作后，就可以使用恢复目录注册数据库。使用恢复目录执行数据库的 RMAN 备份操作前必须注册数据库，这是一个非常简单的进程，在相关的 RMAN 实习中可看到该进程。

RMAN 实习：使用恢复目录注册数据库

实习注意事项

为了完成该 RMAN 实习，应当完成前一个 RMAN 实习(“创建恢复目录”)。

步骤 1　使用 RMAN，同时注册数据库和恢复目录：

```
set ORACLE_SID=main_db
RMAN target=backup_admin/backupuserpassword
CATALOG=rcat_user/rcat_password@recover
```

步骤 2　使用恢复目录注册数据库：

```
RMAN> Register database;
```

步骤 3(可选步骤)　当连接目标数据库时，通过在 RMAN 提示符下执行 report schema 命令来验证是否成功注册数据库：

```
RMAN> Report Schema;
```

---

5. 删除恢复目录

如同可以创建恢复目录模式一样，也可以删除恢复目录模式。使用 drop catalog 命令可以删除恢复目录模式。当然，执行该命令将会使包含在该模式中的所有信息丢失，因此在删除目录模式之前，应该考虑备份恢复目录数据库。

6. 在恢复目录中添加 RMAN 备份

如果已经在没有恢复目录的情况下执行了 RMAN 备份，并且希望在后面将这些备份添加到恢复目录，则可以使用 catalog 命令。我们可以将数据文件副本、备份集片、归档日志备份、甚至整个归档备份目录添加到目录中，如同下面的示例所示：

```
RMAN> CATALOG DATAFILECOPY 'D:\ORACLE\ORA102\DATABASE\system01.dbf';
RMAN> CATALOG ARCHIVELOG 'D:\ORACLE\ORA102\DATABASE\arch_988.arc',
       'D:\ORACLE\ORA102\DATABASE\arch_988.arc';
RMAN> CATALOG BACKUPPIECE 'D:\ORACLE\ORA102\DATABASE\backup_820.bkp';
RMAN> CATALOG START WITH 'D:\ORACLE\ORA102\DATABASE\';
```

注意：
要谨慎使用 catalog start with 命令。必须在目录路径末尾加上反斜杠后缀。如果使用 D:\ORACLE\ORA102\DATABASE，Oracle 将遍历 C:\ORACLE\ORA102 中所有可用的 DATABASE 目录组合。这可能包括如下目录：C:\ORACLE\ORA102\DATABASE，C:\ORACLE\ORA102\DATABASE123，以及 C:\ORACLE\ORA102\DATABASE-OLD。使用反斜杠后缀则指示只需要找到 C:\ORACLE\ORA102\DATABASE\目录。

7. 在恢复目录中取消数据库注册

可以在 RMAN 中使用 unregister database 命令来取消数据库注册。如果希望取消已存在的数据库注册，则只要连接到数据库和恢复目录，并执行 unregister database 命令即可：

```
RMAN>unregister database;
```

如果已经删除数据库，并且希望从恢复目录中删除该数据库，则在大多数情况下只需要知道希望取消注册的数据库的名称。如果希望取消注册 OLDROB 数据库，则可以在连接到恢复目录中执行如下的命令：

```
RMAN>unregister database OLDROB;
```

当恢复目录中有多个以相同名称注册的数据库时，则需要知道希望取消注册的数据库的 DBID。然后在 run 代码块中运行 unregister database 命令，同时使用 set dbid 命令，如同下面的示例所示：

```
rman CATALOG rman/rman@catdb
RMAN> RUN
{  SET DBID 2414555533; # specifies the database by DBID
   UNREGISTER DATABASE ROBOLD NOPROMPT;
}
```

10.1.2　使用虚拟私有目录

在 Oracle 11g 中，可以扩展目录以被多个独立登录的用户使用，从而创建/浏览目录中的记录。在 11g 以前的版本中，所有的人需要使用同样的 RMAN 用户，访问该环境下的所有注册到该目录中的数据库的信息。这种情况将会使单个目录产生安全问题，或由于所有的用户创建并维护自己的目录而导致目录无限扩张。

通过使用虚拟私有目录(VPC)可以使多个用户使用同一个恢复目录。虚拟私有目录允许将对目录的访问权限授予数据库用户，且只能访问该环境下的特定数据库。

为用户构建虚拟私有目录，需要在恢复目录数据库上执行运行在 SQL*Plus 上的操作，也需要在 RMAN 内执行操作。首先，需要登录到数据库，将 recovery_catalog_owner 角色授予使用虚拟私有目录的数据库用户。然后，以目录拥有者的身份登录到 RMAN 并

授予权限——可以是存在的已经注册的数据库的 CATALOG 权限，也可以是 REGISTER 权限，以便用户能够注册到新的数据库上。

除了创建 VPC 外，还可以撤消权限并删除 VPC。

RMAN 实习：创建虚拟私有目录

实习注意事项

在此目录中，将在数据库中为用户 farouka 创建 VPC。目录拥有者为 rcat_user。用户 farouka 需要管理 10g 和 11g 数据库。

步骤 1　在数据库授赋予对目录的访问权限。

```
Sqlplus system/password
SQL> Create user farouka identified by password
Default tablespace catalog_tbs
Quota unlimited on catalog_tbs;
SQL> Grant recovery_catalog_owner to farouka;
```

步骤 2　登录到 RMAN 并授予对数据库 PROD 的访问权限，以及注册的数据库的权限。

```
Rman CATALOG=rcat_user/rcat_password@recover
RMAN> Grant catalog for database PROD to farouka;
RMAN> Grant register database to farouka;
```

步骤 3　退出 RMAN，再次以 farouka 身份登录，然后创建 VPC。

```
Rman CATALOG=farouka/password@recover
RMAN> create virtual catalog;
```

步骤 4　再次登录到数据库中，确保对 10.2 和更早期数据库版本的管理。

```
Sqlplus rcat_user/rcat_password
SQL> execute rcat_user.DBMS_RCVCAT.create_virtual_catalog;
```

10.2　合并多个恢复目录

在使用虚拟私有目录解决了不能充分共享恢复目录的问题后，另外一个令人头痛的问题是消除目录的无限扩展问题。出于某种原因，可能需要停止使用多个目录，在此情况下，需要将一个目录中的记录导入到另外一个目录中，以前使用的方法都会产生巨大的麻烦。

11g 版本的 RMAN 通过使用新的 import catalog 命令提供了一种新的清理合并的方法。可以使用该功能将一个目录中的记录导入到另外一个目录中。使用导入方法，可以

针对源目录中的所有数据库，也可以针对导入执行时所定义的子集。该功能提供了一种对记录进行分类的机制，确保导入正确的记录，不会产生依赖和重复行的问题。

RMAN 的 import 命令可以将两个目录合并。但该命令不能用于升级机制中。在将一个较低版本的目录导入到新的升级版本前，需要首先使用 upgrade 命令对源目录升级，然后才能导入该目录。关于版本问题，应该注意在执行导入命令操作时，源目录、目标目录和 RMAN 可执行文件的版本必须为相同的版本。

import 命令也可以用于将已经存在的目录导入到新的数据库或新的模式中。在这种情况下，应该在新的位置创建新目录，然后使用该新目录作为导入操作时使用的目标目录。

RMAN 实习：将两个恢复目录合并

实习注意事项

本实习将学习将一个存放在目录 RCAT1 数据库的子集导入到目标目录 RCAT2 中。

步骤 1 连接到 RMAN 上的目标目录上：

```
Rman catalog=rcvcat_user/password@RCAT2
```

步骤 2 使通过指定连接到源目录的连接字符串运行 import 命令：

```
RMAN> import catalog rcvcat_user@RCAT1 DB_NAME=TEST, DEV, PROD;
```

步骤 3 确认数据库的元数据导入成功：

```
RMAN> CONNECT TARGET SYS/PASSWORD@TEST
RMAN> list backup of database;
```

10.3 维护恢复目录

使用恢复目录时需要开展一些额外的维护工作，包括在数据库升级或迁移时升级恢复目录，手动重新设置数据库对应物(incarnation)，在执行一些数据库操作后对恢复目录执行重新同步操作。本节主要讨论这些操作，以及其他相关维护问题，包括从恢复目录中删除数据库，使用 Oracle EXP/IMP 实用程序备份恢复目录。最后将概略讲解不同的恢复目录视图及其用途。

10.3.1 取消 RMAN 中的数据库注册

在 Oracle Database 10g 以前的版本中，从恢复目录中取消数据库注册需要手动执行该操作。现在，仅需要执行 unregister database 命令就可以完成该操作。以下是一个实例：

```
RMAN> Unregister database mydb;
```

需要注意的是使用该命令不会将数据库备份文件删除。只是将恢复目录中对这些数据库备份文件的引用删除。还应该注意在执行该命令时需要连接到恢复目录。

10.3.2　数据库迁移/升级问题

在执行 Oracle 数据库升级时，也需要升级恢复目录。正如第 9 章所述，关于所使用的数据库版本、RMAN 版本和恢复目录版本有严格的规则需要遵循。

通过查询恢复目录拥有者模式中 RCVER 视图的 VERSION 列可以确定所使用的恢复目录版本情况。

```
% sqlplus rman/rman @catdb
SQL> SELECT * FROM rcver;
VERSION
----------------------
11.01.00
```

如果表中显示多行，则 RCVER 表中的最高版本号即为当前目录模式版本。例如，假定 RCVER 表显示如下代码行：

```
VERSION
----------------------
08.01.07
09.02.00
10.02.00
```

只要恢复目录的版本与所用的数据库版本同级或者版本更高，就不用担心版本问题。如果需要在同一个恢复目录中存储多个数据库，即使只有一个存储在该恢复目录中的数据库升级，最好还是将该目录升级到高版本。

通过 RMAN 执行 upgrade catalog 命令就可以实现对恢复目录的升级。RMAN 将提示再次输入 upgrade catalog 命令。然后将升级恢复目录。

10.3.3　手动重新设置数据库对应物(reset catalog)

正如前述章节所讨论的情况，每次打开 Oracle 数据库时将会使用 resetlogs 参数，创建一个新的数据库对应物。如果在 RMAN 操作期间执行，则恢复目录将被正确地更新。然而，如果手动执行 resetlogs 命令(例如，通过 SQL*Plus)，则需要重新设置恢复目录中的数据库对应物。该操作可以由 reset database 命令实现：

```
reset database to incarnation 5;
```

10.3.4　手动执行恢复目录的再同步操作(resync catalog)

在 RMAN 使用恢复目录时，可以使用再同步进程以确保恢复目录与目标数据库控制文件保持一致。一般来说，Oracle 在执行备份和恢复这样的 RMAN 操作后，会自动执行数据库再同步操作。所有不必经常对恢复目录执行再同步操作。只有在执行备份时有时使用恢复目录，有时又不使用恢复目录的情况下，则需要对恢复目录执行同步操作。使用 resync catalog 命令可以手动让 Oracle 对恢复目录执行同步操作。

```
resync catalog;
```

Oracle 对恢复目录执行同步操作时，首先将创建一个快照控制文件并将它与恢复目录比较。完成比较后，Oracle 将更新恢复目录使其与数据库控制文件保持同步。

10.3.5 清除恢复目录记录

可能会注意到很少从恢复目录中删除记录。如果不执行维护操作，与其关联的带有 DELETE 状态标志的旧的数据库备份将始终存在于恢复目录中。这种情况将会造成处理备份操作的延时，使备份窗口处于危险中并可能导致会在半夜紧急呼叫 Oracle 技术支持。

如何解决此问题呢？Oracle 提供了一个脚本可以将恢复目录中所有带有 DELETE 标志的记录删除，该脚本位于$ORACLE_HOME/rdbms/admin/prgrmanc.sql 目录中。建议定期使用该脚本以控制恢复目录的大小。

如果想从恢复目录中删除旧数据库的对应物记录，则要从 DBNIC 表中删除这些对应物。可以使用 RC_DATABASE_INCRANATION 视图确定需要删除的对应物。记录想要删除的每个对应物的 DBINC_KEY 值，然后使用 SQL*Plus 通过执行 delete 命令，使用先前记录的 DBINC_KEY 值，从 DBINC 表中删除这些对应物。以下是使用 delete SQL 命令的示例：

```
DELETE FROM dbinc WHERE dbinc_key=2;
```

10.4 备份恢复目录

第 9 章介绍了使用 RMAN 备份数据库的进过程，使用 RMAN 备份恢复目录数据库的进程与之类似。确保创建一个合理的恢复策略，这样才能尽可能快地还原恢复目录。应该记住即使丢失了恢复目录，也并不意味着没有任何解决办法，即使使用恢复目录，也可以在没有恢复目录的情况下恢复数据库。此时，需要数据库控制文件的备份(或者在最坏情况下，可能需要使用 DBMS_BACKUP_RESTORE 做一些技巧性很强的工作)。需要注意的最重要的事情是，要对整个恢复策略进行测试。只有这样才会知道自己的恢复数据库的能力。

10.5 恢复目录视图

恢复目录提供了一系列视图，这些视图可用于探索 RMAN 备份策略产生的元数据。这些视图的数据由注册到数据库时创建的基表提供，当后续执行 resync 命令形成目录时，将会向基表中添加相关的数据。这类视图的命名约定是 RC_*;例如，RC_BACKUP_SET 或 RC_BACKUP_REDOLOG。

RMAN 操作越来越独立于恢复目录，先前唯一针对恢复目录的许多视图被合并到目标数据库控制文件的表中。这意味着对目录中的多数 RC_*视图将会有对应的 v$视图。例如，如果恢复目录视图是 RC_BACKUP_SET，则其 v$视图就是 V$BACKUP_SET。唯一差别在于恢复目录中的主键列在 v$视图中不会存在。

下面将提供恢复目录中关键的 RC_*(以及对应的 v$)视图列表，并给出简要的解释。本书不会给出每个视图的每个列的使用说明，但 Oracle 提供了专门的文档可以用于提供对视图的详细解释。本章将简要介绍什么视图有用，以及如何使用这些视图。因为本书没有介绍代理副本的内容，所以本章忽略了所有与代理副本有关的视图，并忽略了所有与存储脚本有关的视图。

在设置和完善备份策略时，对这些关键视图的访问是非常关键的。毫无疑问，在执行恢复或还原操作前，也会需要访问这些关键视图。将一系列可以按需要运行的脚本汇总到一起是非常有用的，或者能够定期产生一个输出情况报告。下面我们将提供对执行备份和恢复操作非常有用的一些目录查询的实例。

值得注意的是，本章中的所有示例从目标数据库中的 v$视图或者从目录数据库中的 RC_*视图中获得数据。如果通过目录选择并且注册过多个数据库，则应该限制基于 DBID 或者 DB_NAME(若在目录中具有唯一性)的查询。否则将会获得多个数据库的信息，而这些大量的信息没有用处，除非打算收集企业级数据信息。

10.5.1　RC_ARCHIVED_LOG(V$ARCHIVED_LOG)

包含归档日志历史的视图 V$ARCHIVED_LOG 是可以利用的最有用的视图之一。由于混淆了备份归档日志和还原归档日志的特性，对其元数据的访问是非常重要的。记住 RC_ARCHIVED_LOG 不包含归档日志备份的信息(参考 RC_BACKUP_REDOLOG)，仅仅包含由目标数据库创建的归档日志的信息。因为归档日志不断地产生，而目录再同步相对执行频率要低得多，所以经常会使用 V$ARCHIVED_LOG 版本的视图而不是使用 RC_ARCHIVED_LOG 视图。从该视图中获得最多的信息可以参见如下的代码：

```
column name format a50
column completion_time format a25
alter session set nls_date_format= 'DD-MON-YYYY:HH24:MI:SS';
select name, sequence#, status, completion_time
from rc_archived_log;
```

10.5.2　RC_BACKUP_CONTROLFILE(V$BACKUP_DATAFILE)

该视图提供从控制文件中获得的备份信息。这些信息不包括控制文件副本。对使用 copy 命令或使用 catalog 命令创建目录的副本可以使用另外一个视图。若控制文件还原操作异常执行时，该视图可以作为一个极好的参考资源，特别是如果打算为创建备用数据库执行复制操作时。

在 V$BACKUP_DATAFILE 中浏览控制文件的副本，将会查看到文件编号为 0 的记录——表明该文件是控制文件。

```
select file#, creation_time, resetlogs_time,
blocks, block_size, controlfile_type
from v$backup_datafile where file#=0;
```

以下查询将给出数据库 V102 的所有的控制文件备份的信息，这些信息包括结束时间、状态、控制文件类型(普通备份为 B，备用控制文件备份为 S)、自动备份的日期(如果未将控制文件配置为自动备份，则该值为 null)：

```
column completion_time format a25
column autobackup_date format a25
alter session set nls_date_format= 'DD-MON-YYYY:HH24:MI:SS';
select db_name, status, completion_time, controlfile_type,
autobackup_date from rc_backup_controlfile
where db_name= 'V102';
```

10.5.3 RC_BACKUP_CORRUPTION(V$BACKUP_CORRUPTION)

该视图将列出存在于数据文件备份中的损坏。为表示能够容忍的损坏数量，MAXCORRUPT 参数必须设置为非零值，该值指示 RMAN 在产生错误和中止前将备份的损坏块数量。损坏块不会被丢弃，而是被原样备份。

不要将该视图与 RC_DATABASE_CORRUPTION 视图混淆(本章后面将描述该视图)。RC_DATABASE_CORRUPTION 视图将基于数据库执行的最近一次备份操作(或执行 backup validate 命令)列出数据库中的损坏块，RC_BACKUP_CORRUPTION 视图列出的是备份时产生的损坏块而不是数据库本身的损坏块。

下面的代码提供了损坏块列表，包括块编号、文件编号、存在损坏的备份片，以及数据库 V102 包含的损坏类型。

```
select db_name, piece#, file#, block#, blocks, corruption_type
from rc_backup_corruption where db_name='V102';
```

10.5.4 RC_BACKUP_DATAFILE(V$BACKUP_DATAFILE)

该视图包含存在于备份集中的数据文件的大量信息。如果希望查看已经备份的数据文件的特定信息，则可以使用该视图。

10.5.5 RC_BACKUP_FILES(V$BACKUP_FILES)

该视图与 RMAN 的命令行接口的命令 list backup 和 list copy 提供的信息相同。该视图提供了恢复目录包含的所有备份文件的细节，无论该文件是备份集，数据文件副本还是代理副本。

要使用该视图，首先需要调用 DBMS_RCVMAN.SETDATABASE 以指定需要访问的数据库：

```
CALL DBMS_RCVMAN.SETDATABASE(null, null, null, 2283997583, null);
select backup_type, file_type, status, bytes from rc_backup_files;
```

10.5.6　RC_BACKUP_PIECE(V$BACKUP_PIECE)

引用该视图可以获取在一般备份操作期间创建的特定备份片的有关信息。需要注意备份集可包含多个备份片，备份片与备份集逻辑单元的物理文件相似。

10.5.7　RC_BACKUP_REDOLOG(V$BACKUP_REDOLOG)

该视图的名称可能会带来误解：RMAN 不能备份联机重做日志；它只能备份归档重做日志，通常我们将归档重做日志简称为归档日志。该视图将列出存在备份集中于的归档日志。每个备份的归档日志都有一个记录。如果同一个归档日志被备份两次，将会有两个记录存在。

如下查询提供特定范围内的归档日志信息，包括备份集信息，备份集状态和完成时间：

```
alter session set nls_date_format='DD-MON-YYYY:HH24:MI:SS';
select dn_name, bs_key, sequence#, thread#, first_change#, status
from rc_backup_redolog;
```

10.5.8　RC_BACKUP_SET(V$BACKUP_SET)

该视图包含的信息涉及每个逻辑备份集。在使用时需要指定希望查询何种类型的备份集：完整备份、增量备份或者归档日志备份。

10.5.9　RC_BACKUP_SPFILE(V$BACKUP_SPFILE)

利用该视图可以获得存在于备份集中的 SPFILE 备份的信息。

10.5.10　RC_CONTROLFILE_COPY(V$DATAFILE_COPY)

类似于 RC_BACKUP_CONTROLFILE 视图，其对应的视图是 V$DATAFILE_COPY，包含有关编码为 0 的控制文件的信息。在目录中，该视图包含控制文件的副本信息，这些信息包括通过 copy 命令创建的或 catalog 命令目录化的控制文件信息。

10.5.11　RC_COPY_CORRUPTION(V$COPY_CORRUPTION)

该视图与 RC_BACKUP_CORRUPTION 基本相同，不同之处在于该视图报告的是副本中的损坏块，而 RC_BACKUP_CORRUPTION 报告的是备份集中的损坏块。在 select 语句中，将省略 piece#，其他都类似：

```
select db_name, file#, block#, blocks, corruption_type
from rc_COPY_corruption where db_name='V102';
```

10.5.12　RC_DATABASE(V$DATABASE)

该视图包含所有在目录中注册的数据库的基本信息：数据库名、DBID、当前对应物编号、最近的 RESETLOG 时间和 SCN。

10.5.13 RC_DATABASE_BLOCK_CORRUPTION(V$DATABASE_BLOCK_CORRUPTION)

该视图提供了执行 backup 或 backup validate 操作发现的损坏块的信息的列表。需要注意的是该视图提供的损坏块是数据库中实际存在的损坏块，而不是在备份或者副本中的损坏块。该视图在每次备份操作时刷新，反映的是当前损坏(如果存在的话)。在块介质恢复期间，当需要指定 blockrecover corruption list 时，可以使用 V$DATABASE_BLOCK_CORRUPTION 视图，因此该视图是常用视图。以下代码是使用 select 语句查询该视图的示例：

```
select file#, block#, corruption_type
    from v$database_block_corruption;
```

10.5.14 RC_DATABASE_INCARNATION(V$DATABASE_INCARNATION)

该视图包含在目录中注册的每个数据库的每个对应物的记录。其中最重要的信息是 RESETLOGS 信息，该信息通过定义明确定义每个对应物。以下代码是使用 select 语句查询该视图的示例：

```
select dbid, name, dbinc_key, resetlogs_time, current_incarnation
    from rc_database_incarnation
    where db_key=<database key>
    and dbinc_key=<current incarnation key number>;
```

10.5.15 RC_DATAFILE(V$DATAFILE)

该视图的存在使目录能够访问如控制文件所在的位置和数据库中数据文件的详细信息等概略信息。当希望查询有关数据文件的信息时，通常可以使用 V$DATAFILE 视图。然而在恢复期间，如果当前的控制文件不可用，则该视图非常有用。除了数据文件信息外，该视图还包含有关表空间的信息，这一点类似于固定视图 DBA_DATA_FILES。另外，该视图包含命令 configure exclude 和 configure auxname 需要的持久配置信息。以下代码是使用 select 语句查询该视图的示例：

```
select db_name, ts#, tablespace_name, file#, name,
    bytes, included_in_database_backup, aux_name
    from rc_datafile
    where db_name='v102';
```

10.5.16 RC_DATAFILE_COPY(V$DATAFILE_COPY)

该视图包含由 copy 命令创建的数据文件副本或者由 catalog 命令注册的 OS 副本的元数据。

10.5.17 RC_LOG_HISTORY(V$LOG_HISTORY)

V$LOG_HISTORY 视图包含联机重做日志的历史信息，例如日志切换时，以及当 SCN

处于切换状态时。与 V$ARCHIVED_LOG 相比有些冗余的信息，但 V$LOG_HISTORY 并不关心任何当前文件，它只关注日志切换的历史信息。

10.5.18　RC_OFFLINE_RANGE(V$OFFLINE_RANGE)

脱机范围用于设置数据文件脱机或处于只读状态，以及返回至读/写模式(如果存在)时的参数。对 RMAN 来说，备份或还原知道关于一个文件时的这些信息非常重要。从可恢复性角度考虑，知道一个文件处于脱机状态的整个时间范围至关重要。如果在从联机到脱机(或者只读)的过渡之前，存在一个数据文件的备份，则从文件进入脱机状态或者只读状态时开始，直到当前时间都需要归档日志。

10.5.19　RC_REDO_LOG(VLOG, VLOGFILE)

简单来说，类似于 RMAN 需要知道 V$DATAFILE 中关于重建的信息，该视图需要知道联机重做日志文件位于何处。该视图是 V$LOG 与 V$LOGFILE 的组合视图，因此在每个日志名称旁边可获得线程和组成员的信息。

10.5.20　RC_REDO_THREAD(V$THREAD)

在 RAC 环境中，线程信息是最为重要的信息。当重做创建时会立刻产生多个单线程。该视图将当前数据库对应物中每个不同的线程作为一条记录列出，同时列出线程的状态及其重做流。以下代码是使用 select 语句查询该视图的示例：

```
select db_name, thread#,status, sequence# from rc_redo_thread
    where db_name='v102';
```

10.5.21　RC_RESYNC

该视图为产生的目录再同步操作提供信息。显然该视图没有对应的 v$view 视图。利用该视图可以确定您的任何企业数据库是否需要再同步，或者用于诊断可能存在的再同步问题。以下代码是使用 select 语句查询该视图的示例：

```
select db_name, controlfile_time, controlfile_sequence#,
    resync_type, resync_time
    from rc_resync where db_name='V102';
```

10.5.22　RC_RMAN_CONFIGURATION(V$RMAN_CONFIGURATION)

该视图与 show all 命令相同，给出为目标数据库设置的每个配置参数的名称和值。值得注意的是有 3 个配置参数不在此存储：configure exclude 参数的信息可以在 RC_TABLESPACE(V$TABLESPACE)视图中获取；configure auxname 参数的信息可以从 RC_DATAFILE(V$DATAFILE)视图中获取；configure snapshot controlfile 参数的信息只能从目标数据库的控制文件(没有对应的目录)中获取。

应该注意 RC_RMAN_CONFIGURATION 视图没有 DB_NAME 列，因此需要使用 RC_DATABASE 中的主键 DB_KEY 值获得在目录中注册的特定数据库的值。

此外，在 RC_RMAN_CONFIGURATION 视图或 V$RMAN_CONFIGURATION 视图中都没有列出默认值。只有经过手动修改的值才能在列表中查询到。以下代码是使用 select 语句查询该视图的示例：

```
select name, value from rc_rman_configuration where db_key=1;
```

10.5.23　RC_TABLESPACE(V$TABLESPACE)

该视图包含表空间的信息，该视图中相对 V$ TABLESPACE 视图真正有用的信息是保持在目录中的被删除的表空间的历史信息。可以使用该视图回溯并查看表空间是在何时被删除。另外，该视图还包含记录执行 configure exclude 命令的信息。

10.5.24　RC_TEMPFILE(V$TEMPFILE)

版本 10g 以后的 RMAN 能够在还原时追溯并重建临时文件，消除了 10g 版本之前手动实现该操作的困难。RC_TEMPFILE 提供了实现该功能的桥梁，提供了一个了解数据库曾经存在的临时文件的一个窗口。

10.5.25　为 Oracle 企业管理器提供的目录视图

在恢复目录中专门创建了用于增强 OEM 控制台的性能和功能的一系列新视图，并向最终用户提供了有限的功能。这类视图多数无法与目标数据库控制文件中的 v$views 视图对应。为避免误用，有必要浏览这些视图并加以区分。如果希望利用这些视图中的信息，可以发现它们之中存在与 OEM 的备份和恢复功能相似的信息。表 10-2 列出并简单描述了主要为 OEM 使用创建的 RC_*视图。

表 10-2　RC_*视图列表信息

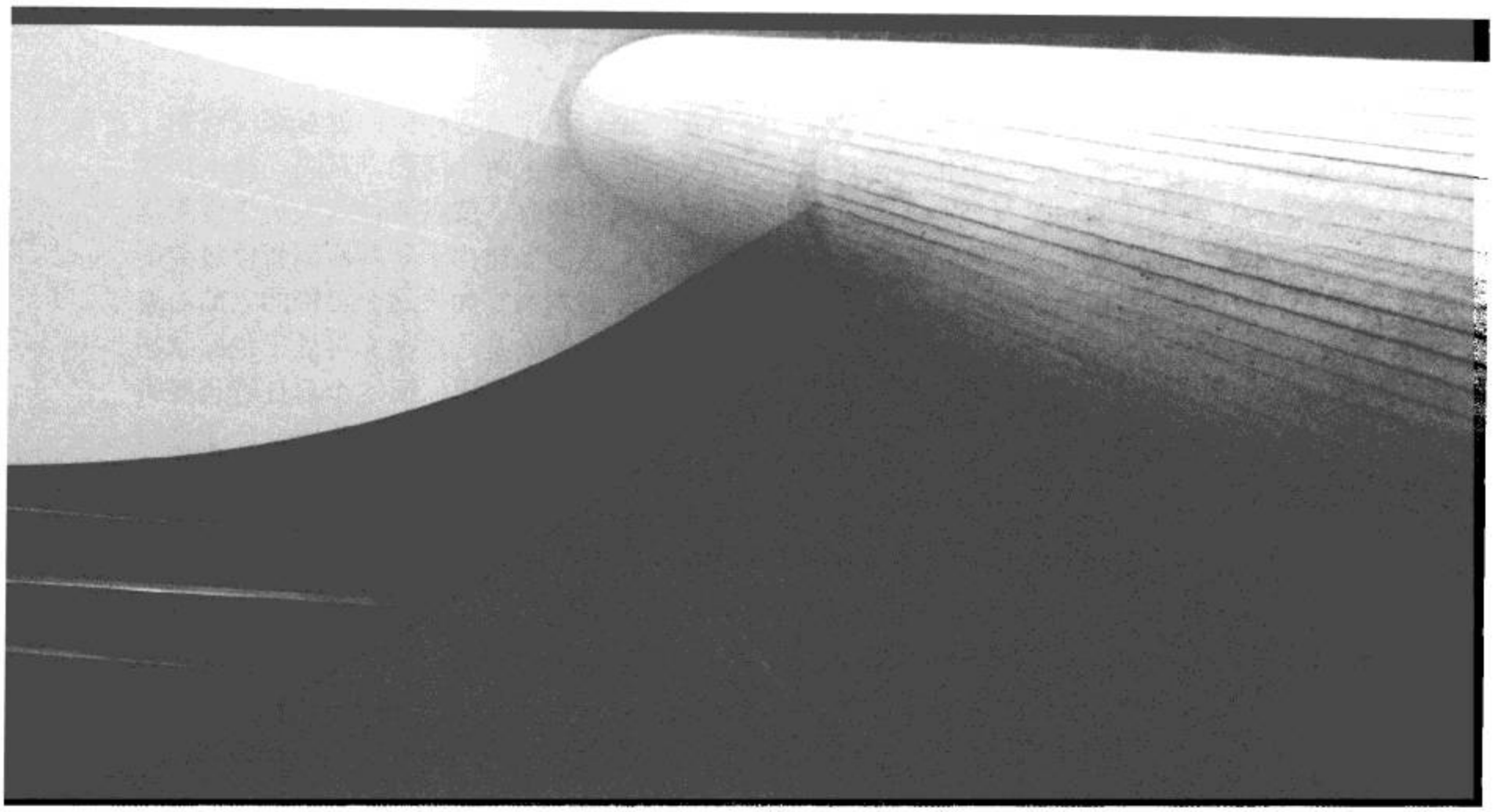
RC_* 视图	描　述
RC_BACKUP_ARCHIVELOG_DETAILS	有关备份的归档日志的详细信息
RC_BACKUP_ARCHIVELOG_SUMMARY	汇总的归档日志备份的信息
RC_BACKUP_CONTROLFILE_DETAILS	控制文件备份的详细信息
RC_BACKUP_CONTROLFILE_SUMMARY	RMAN 知道的所有控制文件备份的汇总信息
RC_BACKUP_COPY_DETAILS	有关所有控制文件和数据文件副本的详细信息
RC_BACKUP_COPY_SUMMARY	有关所有控制文件和数据文件副本的汇总信息
RC_BACKUP_DATAFILE_DETAILS	备份集及映像副本中有关所有数据文件备份的详细信息
RC_BACKUP_DATAFILE_SUMMARY	有关所有数据文件备份的汇总信息
RC_BACKUP_PIECE_DETAILS	目录中可用备份片的详细信息
RC_BACKUP_SET_DETAILS	目录中可用备份集的详细信息——包括使用 backup backupset 命令创建的备份信息

(续表)

RC_* 视图	描　　述
RC_BACKUP_SET_SUMMARY	可用备份集的聚集信息
RC_BACKUP_SPFILE_DETAILS	可用 SPFILE 备份的详细信息
RC_BACKUP_SPFILE_SUMMARY	可用 SPFILE 备份的汇总信息
RC_RMAN_OUTPUT	辅助 OEM 跟踪工作状态，对应的 v$view 视图为 V$RMAN_OUTPUT
RC_RMAN_BACKUP_JOB_DETAILS	单个的备份任务会话，包括会话期间的所有操作的详细信息
RC_RMAN_BACKUP_SUBJOB_DETAILS	RMAN 会话期间，执行类似操作的组的详细信息
RC_RMAN_STATUS	恢复目录中所有的数据库的 RMAN 会话的历史视图。该视图不包括当前会话的信息，对应的 v$view 视图为 V$RMAN_STATUS
RC_UNUSABLE_BACKUPFILE_DETAILS	被标识为 UNAVAILABLE 和 EXPIRED 的任何类型的备份文件列表
RC_RMAN_BACKUP_TYPE	在报告构建期间，为 OEM 提供过滤信息

10.6　小结

本章详细讨论了什么是恢复目录以及如何利用它管理备份，以减少花费在恢复上的时间和精力。本章讨论了如何构建目录，将管理的数据库添加到恢复目录以及如何删除恢复目录。Oracle Database 11g 提供了用于创建和管理虚拟私有目录以维护隐私和安全的选项。另外，11g 还提供了合并多个目录的功能以满足集成操作的需求并简化了对生态系统的管理。最后，提供了一系列重要的恢复目录视图的简要介绍，这些视图可用于理解涉及备份的元数据，帮助指导备份维护操作和恢复操作计划的制定。



第11章

RMAN 备 份

我们已经介绍了启动 RMAN 的所有要素，现在开始使用 RMAN 来实际地执行备份操作。本章介绍使用 RMAN 执行备份操作的全部内容，这些内容包括脱机备份、联机备份、归档重做日志备份以及增量备份等。我们还将介绍如何备份整个数据库和单独的数据库数据文件。

11.1 RMAN 备份优于脚本化备份

为什么要使用 RMAN 来备份数据库？您可能正在使用自己编写的一些很不错的脚本执

行联机备份操作，也许会产生这样的疑惑："在能够可靠地运行脚本时，为什么非要使用RMAN 来备份数据库呢？"我们将在下面解答这个问题。

我们当然相信您备份所使用的脚本不会出现问题。不过请您相信：就算您没问题，但别人写的其他脚本也有可能会出现中断。这种中断会带来两个问题。首先，脚本出现中断时，数据库的备份操作会失败；其次，脚本出现中断时，就需要技术人员来修复这个脚本。另一种情况是，您可能擅长编写 Unix 脚本，然而当您由于某种原因暂停目前的工作后，能保证接任您工作的人也同样擅长编写 Unix 脚本吗？如果这个可怜的人面对您所编写的代码不知所措，那么老板自然会不高兴，更重要的是，数据库将处于危险状态中。当然，也有可能是由您来调试前任者留下的令人费解的代码，而这个前任者已经去空间站工作了！而这正是 RMAN 的优势所在：它得到了 Oracle 的支持。因此，我们可以使用 Oracle 的 RMAN 来执行备份操作，以摆脱上述的烦恼。

当然，使用 Oracle Database 11g 中的 RMAN 执行备份操作还有其他许多优点，这些优点包括：

- RMAN 会检测和报告损坏的数据块。
- 不需要将表空间置入热备份模式，RMAN 就可以联机备份数据库。因此，热备份期间额外生成的重做(有时非常多)会减少。
- RMAN 会自动跟踪新的数据文件和表空间，这样就不再需要在脚本中添加新的表空间和数据文件。
- RMAN 只备份使用过的数据块(直至最高使用标记(high water mark，HWM))，这样 RMAN 备份映像通常小于联机备份脚本的备份映像。
- RMAN 提供备份映像的实际压缩。
- RMAN 提供自动且方便的备份、还原和恢复操作。RMAN 跟踪恢复数据库所需的所有备份，如果在要求还原操作时需要使用这些备份，并且只还原必需的对象。
- RMAN 可以与第三方介质管理产品一起无缝地工作。
- RMAN 支持增量备份策略。
- 可以使用 RMAN 来测试备份而不需要实际还原它们。使用备份脚本是无法实现这个功能的。
- 如果使用存储库，RMAN 会提供一个不错的集中化报告工具。
- 如果运行 Oracle Database 11g，新的数据恢复顾问(Data Recovery Advisor，DRA)可以快速地诊断数据库恢复问题。并提供还原或恢复建议并可通过 RMAN 自动还原。
- 使用 RMAN 的 DRA 可简化对困难问题的诊断工作。利用 11g 的 RMAN 的 ADR 命令解决发现的问题。

如果使用过 Oracle Database 10g 之前版本中的 RMAN，将发现早期的 RMAN 备份命令仍然有效。RMAN 向后兼容。然而，如果没有花费时间来回顾 Oracle Database 11g 中的 RMAN 提供的特性，并且没有实现这些有用特性，您就不会获得 RMAN 的所有优点。

11.2　RMAN 兼容性问题

在启动备份操作之前，需要考虑一些兼容性问题。在一个企业中，可能会运行不同的 Oracle 版本，因此我们在设计备份策略时必须要考虑 RMAN 的兼容性问题。并不是所有数据库都能兼容所有的 RMAN 版本，如果再加上恢复目录，情况就会更复杂。表 11-1 提供了 RMAN 兼容性问题的快速参考。

在表 11-1 中，可以看到 RMAN 目标数据库版本(假定数据库版本是 10.2.0)和支持该数据库版本备份的 RMAN 客户端(在示例中，10.2.0 数据库可以被版本≥9.0.1.3 并且直到 10.2.0 版本的 RMAN 备份)。同样，可以看到 RMAN 目录数据库的版本必须能够支持该数据库的备份(例如对于版本为 10.2.0 的数据库，应使用 9.0.1 版本的目录数据库)。最后，列出了必要的目录模式的版本(≥在示例中使用的 RMAN 客户端版本)。

表 11-1　RMAN 兼容性矩阵

RMAN 目标数据库版本(应用补丁)	RMAN 客户端版本(应用补丁)	RMAN 目录数据库版本(应用补丁)	RMAN 目录模式版本(应用补丁)
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	≥RMAN 客户端
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	≥RMAN 客户端
9.2.0	≥9.0.1.3 并且≤目标数据库可执行文件版本	≥8.1.7	≥RMAN 客户端
10.1.0	≥9.0.1.3 并且≤目标数据库可执行文件版本	≥9.0.1	≥RMAN 客户端
10.2.0	≥9.0.1.3 并且≤目标数据库可执行文件版本	≥9.0.1	≥RMAN 客户端
11.1.0	≥9.0.1.3 并且≤目标数据库可执行文件版本	≥9.0.1	≥RMAN 客户端
11.2.0	≥9.0.1.3 并且≤目标数据库可执行文件版本	≥9.0.1	≥RMAN 客户端

从表 11-1 中可以看到，需要知道恢复目录模式的版本。恢复目录模式中的 RCVER 版本将提供该信息。下面是相关示例。

```
SQL> select * from rcver;
VERSION
------------
10.02.00.00
```

最后，如果要创建多个恢复目录，只要数据库为 9.0.1 或更高的版本，我们就不需要担心在同一个数据库中不能保留所有恢复目录的问题。这个问题仍然是由单独的一个恢复目录数据库解决的，我们可以从这个数据库中方便地得到企业范围的报告。

11.3 监控 RMAN 备份状态

RMAN 在备份执行期间产生输出。如果在启动 RMAN 时启用登录，则该输出将被取消。可以通过查看生成的日志文件来监控 RMAN 操作，或者使用 V$视图的 V$RMAN_OUTPUT 监控 RMAN 操作，如下所示：

```
SQL> select output from v$rman_output order by stamp;
OUTPUT
----------------------------------------------------------------Starting
backup at 12-NOV-05
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=138 devtype=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: sid=154 devtype=DISK
channel ORA_DISK_1: starting compressed full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00001
name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSTEM01.DBF
input datafile fno=00004
name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 12-NOV-05
input datafile fno=00003
name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSAUX01.DBF
channel ORA_DISK_2: specifying datafile(s) in backupset
channel ORA_DISK_2: starting compressed full datafile backupset
input datafile fno=00005name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\EXAMPLE01
.DBF
input datafile fno=00002name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\
NDOTBS01.DBF
channel ORA_DISK_2: starting piece 1 at 12-NOV-05
```

11.4 脱机 RMAN 数据库备份

前面介绍了许多与 RMAN 备份相关的内容，这里我们要讨论的第一个备份主题是执行 Oracle 数据库的脱机备份(或冷备份)操作。显而易见，执行脱机 RMAN 备份操作时只

加载了数据库，但是没有打开数据库。如果按照第 3 章所述的步骤为 RMAN 配置了默认的设置，那么执行脱机 RMAN 备份操作就非常简单了。

11.4.1　使用默认的设置执行脱机备份操作

要执行脱机备份操作，首先需要登录 RMAN(在本章所提供的脱机备份操作示例中，没有使用恢复目录)。接下来，使用 RMAN 命令 shutdown 和 startup mount 来加载数据库，执行脱机备份操作时数据库必须处于这种状态。一旦加载了数据库，就可以简单地通过 backup database 命令来执行脱机备份操作。下面是一个通过 RMAN 执行脱机备份操作的命令示例：

```
shutdown
startup mount
backup database;
startup
```

如果愿意，还可以将数据库作为压缩的备份集来备份：

```
shutdown
startup mount
backup as compressed backupset database;
startup
```

RMAN 实习：执行脱机备份操作

实习注意事项

本实习假定已经为数据库配置了第 3 章所介绍过的自动信道，且为备份操作配置了一个名为 backup_admin 的数据库账户(第 3 章也曾做过介绍)。此外还假定使用已配置的 MML 层。

步骤 1　启动 RMAN:

```
C:\>rman target=backup_admin/robert
```

步骤 2　使用 shutdown immediate 命令关闭数据库:

```
RMAN> shutdown immediate
```

步骤 3　使用 startup mount 命令加载数据库:

```
RMAN> startup mount
```

步骤 4　使用 backup database 命令备份这个数据库。在这种情况下，为了节省磁盘空间，将压缩备份集(因为没有作为默认设置配置压缩):

```
RMAN> backup as compressed backupset database;
```

步骤 5　使用 alter database open 命令打开这个数据库。

```
RMAN> alter database open;
```

下面是一个按照这些步骤执行的脱机 RMAN 备份的完整示例:

```
C:\>rman target=backup_admin/Robert
RMAN> shutdown
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area 272629760 bytes
Fixed Size                  1248504 bytes
Variable Size              83886856 bytes
Database Buffers          184549376 bytes
Redo Buffers                2945024 bytes
RMAN> backup as compressed backupset database;
Starting backup at 04-NOV-05
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=157 devtype=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: sid=155 devtype=DISK
channel ORA_DISK_1: starting compressed full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00001
name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSTEM01.DBF
input datafile fno=00004
name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 04-NOV-05
channel ORA_DISK_2: starting compressed full datafile backupset
channel ORA_DISK_2: specifying datafile(s) in backupset
input datafile fno=00003
name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSAUX01.DBF
input datafile fno=00005name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\
EXAMPLE01.DBF
input datafile fno=00002name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\
UNDOTBS01.DBF
channel ORA_DISK_2: starting piece 1 at 04-NOV-05
channel ORA_DISK_1: finished piece 1 at 04-NOV-05
piece handle=
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2\BACKUPSET\2005_11_04\
O1_MF_NNNDF_TAG20051104T102913_1PQ32XLB_.BKP
tag=TAG20051104T102913 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:12
channel ORA_DISK_2: finished piece 1 at 04-NOV-05
piece handle=
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2\BACKUPSET\2005_11_04\
```

```
O1_MF_NNNDF_TAG20051104T102913_1PQ33J52_.BKP
tag=TAG20051104T102913 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time: 00:01:11
Finished backup at 04-NOV-05

Starting Control File and SPFILE Autobackup at 04-NOV-05
piece handle=
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2\AUTOBACKUP\2005_11_04\
O1_MF_S_573474457_1PQ357T0_.BKP comment=NONE
Finished Control File and SPFILE Autobackup at 04-NOV-05
Finished Control File and SPFILE Autobackup at 04-NOV-05
RMAN> alter database open;
```

可以看到，在前面的示例和 RMAN 实习中，我们只使用了少量的命令。由于已经定义了默认的配置设置(参阅第 3 章)，因此 RMAN 会自动使用默认的设置。除了执行 shutdown 和 startup mount 命令关闭和重新启动数据库，不需要再执行其他操作。然后执行 backup as compressed backupset database 命令进行备份。是不是非常容易？于是 RMAN 就备份数据库数据文件、控制文件和服务器参数文件(假定已经配置了这些功能)。一旦备份了这些文件，就只需要执行 alter database open 命令打开数据库，同时结束备份操作即可。

在这个示例中，Oracle 创建了两个备份集，每个备份集含有一个备份片。从下面的输出中可以看到，这些备份片将被创建在数据库的闪回恢复区(FRA)中，即 C:\ORACLE\PRODCUT\10.2.0\FLASH_RECOVERY_AREA：

```
piece handle=
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2\BACKUPSET\2005_11_04\
O1_MF_NNNDF_TAG20051104T102913_1PQ33J52_.BKP
tag=TAG20051104T102913 comment=NONE
```

注意：
Oracle 只支持 SPFILE 的备份。使用 RMAN 不可以备份数据库中基于文本的 init.ora 参数文件。

最后，我们可以在执行这个备份操作时选择连接恢复目录(本章中执行的所有备份操作都可连接恢复目录)。要连接恢复目录，只需要在启动 RMAN 时添加 catalog 参数即可：

```
C:\>set oracle_sid=recover
C:\>rman target=backup_admin/Robert catalog=rcat_owner/password@robt
```

这里需要注意的是，连接恢复目录属主时使用的是 Oracle Net，这是因为我们将 ORACLE_SID 设置为数据库的 SID，而不是设置为恢复目录的 SID。使用恢复目录执行备份操作时，需要使用一个服务器名和 ORACLE Net 来连接恢复目录或要备份的数据库。通常，建议使用网络连接来连接恢复目录，并尽可能直接地连接数据库。

另外需要注意的是，如果没有配置控制文件的自动备份操作，只要备份数据文件 1，RMAN 仍然会备份控制文件，此时控制文件会被备份在数据文件 1 所在的备份集中。

在结束数据库备份操作之后，还要执行单独的控制文件备份操作，这样可以备份最新的控制文件(这是因为在备份集中备份的控制文件不会包含当前备份的完整信息)。需要注意的是，如果必须要恢复此控制文件，其操作比配置控制文件自动备份情况下所执行的恢复操作更为复杂，所以强烈建议您在系统上配置控制文件的自动备份。

11.4.2 不使用默认的设置执行脱机备份操作

如果没有配置默认设置(参阅第 3 章)，会出现什么情况？如果默认设置不符合要求(我们可能不希望备份到 FRA)，又会出现什么情况？这时，我们需要执行一些其他操作。下面的示例说明了一个这样的备份，我们可以从这个示例确定要执行的操作：

```
shutdown
startup mount
run
{
allocate channel c1 device type disk format 'd:\backup\robt\robt_%U';
allocate channel c2 device type disk format 'c:\backup\robt\robt_%U';
backup as compressed backupset database;
backup current controlfile;
}
```

下面是一些具体步骤。

1. 首先关闭和加载数据库

这个示例比前面的示例复杂一些。首先，我们也会使用前一个示例中的 shutdown 和 startup mount 命令。这些命令是任何脱机备份操作所必需的命令。我们将在本章后面讨论联机备份。

2. 运行 Oracle

接下来，要使用一个 run 代码块，这个代码块是位于 run 命令的花括号中的一组语句，这些语句作为一个代码块一起执行。在完成整个语句块的输入之前，Oracle 不会运行任何语句。一旦输入了所有语句，使用结束花括号可以完成 run 代码块的编写。一旦输入了结束花括号(当然紧接着要按 Enter 键)，就会编译并执行 run 代码块。

注意：

本书是使用 Oracle Database 11g Release 2 编写的。在这个 RMAN 版本中(以及 9*i* 和 10g 版本)，run 代码块中不再需要以前运行的许多命令。如果这个版本不作要求，我们就不会使用 run 代码块。在本章及后面章节中介绍的许多备份和还原/恢复命令在以前的版本中仍可以运行，但需要在 run 代码块中运行。

3. 分配信道

在前面的代码示例中，run 代码块中含有多个不同的 RMAN 命令。我们会先执行 allocate channel 命令，每个命令为 RMAN 分配一个信道来执行数据库备份操作。第 3 章已经介绍过信道的内容，现在再介绍一些这方面的知识。

首先，将介绍备份集和备份集片。每次创建一个信道时，都会创建一个或多个备份集。但也会存在一些例外情况，不过通常是这样。为了便于讨论，我们假设这句话是正确的。下面简要地定义了一些术语：

- **备份集(backup set)**：逻辑实体，我们会为每个定义的信道创建一个或多个备份集(通常为每个信道创建一个备份集)。
- **备份片(backup piece)**：驻留备份数据的实际物理文件。每个备份集可能关联一个或多个备份片。

可以使用 backup 命令(或者配置一个默认值)来控制整个备份集的大小，也可以使用 allocate channel 命令(或者在配置默认信道时配置一个默认值)来控制整个备份片的大小。本章后面部分将进一步介绍限定备份集大小的问题。

allocate channel 命令定义给定信道(进而是一个备份集)所分配到的设备。设备可以是磁盘(type disk)，也可以是磁带驱动器(type sbt)。如果为磁带系统分配信道，可能要在这条命令中加入正在使用的 MML 供应商所要求的特定参数设置。下面是使用 MML 供应商 VERITAS NetBackup 为磁带分配信道的 allocate channel 命令示例：

```
allocate channel t1 type sbt parms='ENV=(NB_ORA_CLASS=RMAN_db01)';
```

执行这条命令可以为一个磁带设备分配一个信道(参阅第 4~10 关于非磁盘备份位置的主题章，可以了解为 MML 设备，Amazon S3 及 Oracle Secure Backup 分配 RMAN 信道的更多内容)。为备份操作分配完两个信道后，RMAN 会在这两个信道间自动并行处理备份流。由于分配了两个信道，因此需要创建两个备份集，并且每个备份集拥有一个备份片。备份片的大小按字节计算，不过也可以根据需要使用 k、m 或 g 占位符来分别表示千字节、兆字节或十亿字节。下面是 allocate channel 命令的另一个示例：

```
allocate channel t1 type disk maxpiecesize=100m;
```

在这个示例中，我们限定了通过 t1 信道创建的备份集中每个备份片的大小不超过 100MB(始终牢记每个信道将创建单个备份集)。这样能够最好地保证所创建的单个备份片大小不会超出磁带或文件系统的处理范围。下面是 allocate channel 命令的另一个示例：

```
allocate channel t1 type disk maxpiecesize=100m
    format='d:\backup\robt\robt_%U.bak'
```

在这个示例中，使用了 format 参数来定义备份片放入磁盘的位置以及备份片的命名约定。由于使用 format 参数，因此需要告知 RMAN 不使用备份集片的默认位置(通常是 FRA)。这里需要注意的是 format 命令中的%U 格式占位符，由于不会备份到 FRA 中，因此需要为将创建的备份片定义关联的文件命名约定。在该示例中使用%U 格式指示符。

执行这条命令得到的备份片名将包括数据库名称(这里是 robt)，且后跟一个下划线和一个助记符。这个包含 8 个字符的助记符由下面的内容组成：

- 备份集标识符。创建备份集时，RMAN 会为每个备份集指派一个唯一的标识码。
- 创建备份集片的时间。

在助记符后面是一个下划线，然后是备份集片号。备份集片号唯一地标识备份集中的每个备份片，因此对于一个备份集来说，这个备份集片号是唯一的。备份集片号后面又是一个下划线，最后是备份集片的副本号。每个多路复用的备份集片副本也具有唯一的副本号。如果没有使用多路复用技术，副本号就是 1。

下面是一个备份集片名的示例：

```
Rob1_16E112V9_1_1
```

在这个备份集片名中，八位助记符的时间组件不容易被辨别，不过这并不重要。使用%U 占位符最重要的是保证每个备份集片的名称是唯一的。当然，format 命令可以使用多个不同的助记符，不过通常只需要使用%U 占位符。出于良好的习惯，最好在这个备份集片名中添加实例名和扩展名。最后，allocate channel 命令还有其他一些用法，参阅附录 A 可以得到 allocate channel 命令的整个语法。

如果使用 FRA，Oracle 将会基于 OMF 命名标准创建备份集文件名。参考 *Oracle Database 11g Release 2 Database Adminrstrators Guide*，可获得关于 Oracle OMF 命名标准的更多详细内容。

注意：

我们使用 SBT 来代替 SBT_TAPE。RMAN 的早期版本使用 SBT_TAPE，而现在只使用 SBT。不过为了实现向下兼容性，SBT_TAPE 仍然是可用的。

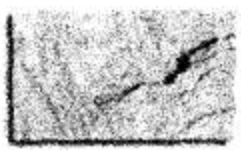
信道可能会失效，特定的硬件也可能会失效，或者会发生其他一些故障。许多 RMAN 备份由多个进入不同位置的信道组成。如果 RMAN 正在使用多个信道但一个信道失效，剩余的信道就会尝试完成失效信道的工作。

4. 备份操作

在示例代码中，分配信道之后，就可以使用 backup 命令来备份数据库(使用 database 选项)。执行 backup database 命令的结果是 RMAN 开始使用所创建的两个信道备份数据库，这条命令与前面执行的 backup database 命令稍有不同，这里的 backup database 命令被限定在 run 命令代码块中。由于使用 allocate channel 命令手动分配信道，所以必须使用 run 代码块来执行这个备份操作。

默认情况下，如果备份数据文件 1(在脱机备份或完整备份操作期间通常会备份该文件)，backup 命令也会备份控制文件和服务器参数文件(SPFILE)。在这种情况下，控制文件备份的存储位置取决于 controlfile autobackup 参数的设置。如果 controlfile autobackup 参数值被设置为 off，控制文件和服务器参数文件(如果使用了服务器参数文件)会被包含在数据库备份集中；如果 controlfile autobackup 参数值被设置为 on，控制文件和服务器

参数文件备份就会生成到一个独立的控制文件备份片。可以在 backup database 命令中加入 include current controlfile 子句，迫使 RAMN 将控制文件放入数据库备份集(假设不备份数据文件 1)。如示例所示，最好单独备份控制文件，以确保备份的是当前的控制文件，因为其中包含了最新的数据库备份。



注意：
RMAN 只备份服务器参数文件(SPFILE)，而不会备份基于文本的 init.ora 文件。

backup database 命令还有许多不同的选项(实际上该命令也是 backup 命令的一个子集)，下面将介绍一些 backup 命令选项的用法。

注意：
Oracle 11g 的 RMAN 备份提供了一个新的特性，该特性消除了对数据库内多数 UNDO 操作的备份。因为在数据库恢复期间多数的 UNDO 操作不需要恢复(因为涉及的事务已经提交过)，所以不需要对其进行备份。该特性可以大大减少 UNDO 表空间的备份容量。注意该特性不能被设置为禁用。

11.5　Backup 命令选项

现在将介绍不同 backup 命令选项的用法，这些命令选项可以构成多种 backup 命令，例如 backup database(上一节刚作过介绍)、backup tablespace、backup datafile 等，本章后面将介绍这些选项。我们可以使用 backup 命令的不同选项为单个备份片提供命名格式或限定这些备份片的大小，如果想重写 RMAN 的选择，甚至可以使用 backup 命令手动地设定使用什么信道备份什么文件。下面将介绍 backup 命令的这些选项。

11.5.1　多段备份

Oracle Database 11g 的一个新特性是可以将大的数据文件备份划分为固定大小的多段备份。可以使用不同的信道备份这些段，并行化大型数据库数据文件的备份操作。如果使用了 Oracle 10g 推出的 bigfile 表空间，则这一新特性将会很有帮助。利用该新特性进行的备份称为多段备份。

为实现多段备份，需要在 RMAN 中指定段的容量参数。RMAN 将把该待备份的文件划分为文件块，该划分是将文件逻辑划分成连续块。RMAN 将创建备份集，每个文件段使用一个备份集片。以下示例为名为 USER_DATA 的 bigfile 表空间备份，并将备份段设置为 1GB：

```
backup section size 1g tablesapce USER_DATA
```

多段备份将备份负载分散到多个不同的 I/O 设备中，使备份并行化。是一个特别有

效的新特性。

11.5.2 压缩

在前面的示例中可以看到，我们可以实际地压缩备份集。默认情况下，RMAN 执行 NULL 数据块压缩。RMAN 也提供了备份集的实际压缩，这种压缩可以减少备份映像所占用的全部存储空间。下面讨论这两种不同类型的压缩。

1. NULL 数据库块压缩

使用这种形式的压缩时，Oracle 不会备份未使用的数据块。执行 NULL 数据块压缩的情形有以下两种：

- 从未使用过的数据块不被备份。
- 在给定的特定标准下使用过一次的数据块也不被备份。

在第一种情况下，将不会备份其中一直没有数据的所有数据块。在第二种情况下，如果数据库和关联的数据块满足某种标准，空块将不会被备份，即使它在某个时刻包含数据。下面是允许 RMAN 不备份这些数据块所必须满足的条件：

- 将兼容参数设置为 10.2。
- 没有为数据库定义任何有保证的还原点。
- 本地管理数据文件。
- 备份是备份集，并且是完整备份或 0 级增量备份。
- 备份集在磁盘上创建。

如果满足这些条件，Oracle 将不会备份任何未使用的块，并且备份将因此占用磁盘或磁带上的较少空间。

2. RMAN 备份压缩

在本章的前面提供了配置数据库以使用 RMAN 压缩的示例。RMAN 可以将压缩算法应用于备份集，从而减小 RMAN 创建的备份集的大小。RMAN 压缩可以在很大程度上减小备份集的大小。有时，压缩可能非常有效；例如，在测试数据库中，当使用压缩时，备份集映像的大小减少了 70%。如果没有配置数据库以自动压缩备份集，则可以使用 as compressed backupset 参数来创建压缩的备份集。如果已经配置了压缩并且不希望在给定的 backup 命令中使用压缩，则只需要使用 backup 命令的 as backupset 参数(没有 compressed 关键字)即可。

Oracle Database 11g Release 2 的 RMAN 提供如下几种不同的压缩选项供选择：

- 默认压缩
- 低级压缩
- 中级压缩
- 高级压缩

缺省压缩类型与 Oracle Database 10g Release 1 开始提供的压缩方式相同，并且不需要许可。低级、中级和高级压缩方式提供影响整个系统压缩的控制方式。低级压缩方式

提供最小影响 CPU 方面的一些压缩，而中级和高级压缩增量式地提供更好地压缩，从而提供更好的性能。如果使用非默认压缩方式，则需要从 Oracle 购买不同的许可。

可以通过使用 RMAN configure 命令(第 3 章中有相关的讨论)将压缩配置为默认值。以下是在 Oracle 11*g* Release 2 Database 中配置默认压缩的示例：

```
Configure compression algorithm 'DEFAULT';
Configure compression algorithm 'HIGH';
Configure compression algorithm 'MEDIUM';
Configure compression algorithm 'LOW';
Configure compression algorithm clear;
```

也可以在 Oracle Database 10*g* 和 Oracle Database 11*g* Release 1 上配置默认压缩。其配置方式略有不同。请参考相关 Oracle 文档，获取更多关于在不同的数据库版本上确保实现压缩的相关信息。

11.5.3　标记和还原点

在 Oracle 11*g* 及更早期的版本中，标记和还原点提供了类似的功能。都可以将数据库恢复到给定的时间点。在第 14 章中会详细讨论时间点(不完全)恢复。本节将主要讨论标记和还原点。

1. 标记

可以为 Oracle 中的每个备份分配一个标记(tag)。标记是不超过 30 个字符的与特定备份有关联的名称，该名称能在还原操作期间被引用以指示将使用的特定备份。该标记可应用于完整备份、表空间备份、数据文件备份、增量备份甚至备份副本(本章将讨论所有这些备份)。下面是为完整备份分配一个标记的示例：

```
backup database tag='test backup';
```

在这个示例中，需要注意的是使用了 tag 参数来标识这个完整备份。每个标记都应该是唯一的，如果没有指定一个命名约定，RMAN 就会使用默认的命令约定为每个备份集分配一个标记。相同的标记可以应用于多个备份，并且默认情况下只还原最新的备份。

2. 还原点

标记与特定备份相关联时，还原点与特定时间点相关联。RMAN 没有提供创建还原点的命令。可以使用 RMAN 的 sql 命令执行如下 create restore point 的 SQL 命令：

```
SQL "Create restore point Charlie_one";
```

也可以通过 Sql *Plus 时间点提供的 create restore point 命令创建还原点：

```
SQL> Create restore point Charlie_one;
```

如创建备份时使用 keep 选项，则在备份期间也可以创建还原点。在本章以后关于 keep 选项的章节中可以获得更多关于便于 keep 选项创建还原点的信息。

在 RMAN 还原期间可以引用还原点代替其他时间点还原方法(例如基于时间还原)。在第 14 章中将更加详细地讨论使用还原点恢复的问题。

11.5.4 限定备份的影响

为了帮助减少 RMAN 备份对其他进程的整体 I/O 影响，RMAN 提供了 backup 命令的 duration 参数。duration 参数类似于闹钟：如果备份运行时间超过指定的持续时间，RMAN 将取消备份。下面是使用 backup duration 命令的示例：

```
backup duration 00:30 database;
```

关于 duration 参数的一个注意事项是不可以使用 backup database plus archivelog 命令。duration 参数也可用来抑制备份。当定义持续时间时，可以让 RMAN 使以下因素之一最小化：

- 备份运行的时间
- 备份使用的 I/O 负载

如果尝试最小化备份运行的时间，RMAN 将预先全速备份。这是默认的设置。使用默认的 minimize time 参数的另一个特性是，RMAN 将区分备份的数据文件的优先级。最近备份的数据文件具有较低的优先级，而较早备份的数据文件具有较高的优先级。

也可以让 RMAN 尝试将备份的 I/O 操作分布到持续时间内的各个时间点上，从而消除备份对系统的整体影响：

```
backup duration 00:30 minimize time database;
backup duration 00:30 minimize load database;
```

最后，使用 duration 参数，可以指示 RMAN 如何处理超出 backup duration 时间限制的备份。当使用 partial 参数时，如果备份因为超出 duration 参数的持续时间而终止，RMAN 不会将其作为失败的备份进行处理。因此，任何 run 代码块中的剩余命令将继续执行。因此，如果接下来还有 backup 命令，例如归档的重做日志的备份，使用这个参数就会非常方便。无论 partial 参数的设置是什么，即使整个备份过程没有完成，Oracle 也会将已经成功完成的备份集认为是可用的。

11.5.5 限定备份集的大小

在上一个示例的基础上，通过添加新的 maxsetsize 参数，就可以得到下面的示例：

```
backup database maxsetsize=50m tag='test backup';
```

可以使用该参数限定备份集的大小不超过 50MB。如果对磁带的大小有限制，或者磁盘只能处理一定大小的数据文件，使用这个参数将会非常方便。Oracle 将备份分成多个备份集，每个备份集都不会超出所定义的 maxsetsize 参数值。

但 maxsetsize 参数引入的问题是它限定了单个备份集的总体大小，因此，如果备份集中的数据文件大小超出定义的限定范围，备份操作就会失败，下面的示例说明了这种情况(示例中省略了一些输出)。

```
RMAN> set maxcorrupt for datafile 1 to 10;
RMAN> run
2> {
3> allocate channel c1 device type disk format 'd:\backup\robt\robt_%U';
4> allocate channel c2 device type disk format 'd:\backup\robt\robt_%U';
5> backup maxsetsize=50m tag='test backup' database;
6> }
allocated channel: c1
allocated channel: c2
Starting backup at 13-JUN-02
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of backup command at 06/13/2002 22:07:47
RMAN-06183: datafile or datafilecopy larger than SETSIZE: file# 1
D:\ORACLE\ORADATA\ROBT\SYSTEM01.DBF
```

因此，使用这个限定参数时要特别小心，有时候也可以使用 allocate channel 命令的 maxpiecesize 参数来完成这个操作。

此外，还可以使用首选的 configure 命令创建备份集大小的默认限定值和备份片大小的限定值。参阅第 3 章可以得到更多相关的信息。

11.5.6　备份到特定的设备类型

可能会配置不同的默认信道，例如一个到磁盘，一个到磁带。可以使用 device type 参数定义在开始备份时使用哪个自动信道设备。如下例所示：

```
backup database device type disk;

backup database device type sbt;
```

11.5.7　修改备份集的保留策略

从 Oracle 11g 开始，RMAN backup 命令中的 keep 参数用于重写默认保留策略并建立称为归档备份的备份。归档备份是一个自包含的备份(意味着它包含用于建立一致性备份的归档重做日志)。注意从 Oracle 11g 开始，logs 和 nologs 选项不再有效，因为日志始终被备份。keep 参数包含以下选项：

- **forever**：指示归档备份将被永远保存，除非用手动方式删除它。利用 keep forever 选项需要使用恢复目录，因为控制文件记录可能会因为时间过长而不存在。下面为在备份期间使用 keep forever 参数的示例：

```
backup database keep forever;
```

- **until time 字符串**：该选项定义用于备份的可选保留策略。应该注意的是，如果时间超过 365 天，控制文件中的记录将可能消失，无论如何，RMAN 不需要使用恢复目录，因为当使用 keep forever 参数时已经使用。

```
backup database format'c:\oracle\backup\%U' keep untiltime ='sysdate+
180' tag Keep_backup;
```

该方式易于区分用于还原目的的备份。以下的示例将建立归档备份并将其分配到一个称为 gold_copy 的还原点：

```
backup database format 'c:\oracle\backup\%U'
keep until time = 'sysdate+180'
restore point gold_copy;
```

对归档备份的一个限制是它们不能被存储在闪回恢复区中(FRA)中。如果打算使用 FRA 存储归档备份，RMAN 将产生错误，并导致备份出错。

keep 选项在 Oracle 10g 及早期的版本中也可用。但执行起来有些不同。在先前版本的 keep 选项中，需要使用不同的 RMAN 命令备份归档重做日志，因为在使用 keep 选项时没有 plus archivelog 选项。由于归档重做日志不能被自动备份，您实际上有一个不可还原的备份，因此要谨慎操作。过去的方法通过使用 logs 选项将所有归档重做日志备份集标记为使用同样的保留策略，或者跳过标记归档重做日志备份集为 nologs 选项。在 Oracle 11g 中，logs 和 nologs 选项对归档备份不再有效。

如果使用 Oracle Database 10g 及更早版本的 keep 选项，则需要了解以下选项：

- logs：包含归档重做日志的备份的备份集与备份它们时分配的保留策略一致。这样的效果是将与备份同期的日志备份集保存，使一致性恢复成为可能。
- nologs：指示所有归档重做日志将采用同样的保留策略。如果归档重做日志过期，备份尚未过期，将导致 ARCHIVELOG 归档模式备份不可用，该参数对 UNARCHIVELOG 模式备份有效，因为没有需要备份的归档重做日志。

以下给出的示例介绍了如何在 Oracle 10g 中应用 keep 参数。注意使用 plus archivelog 参数确保将归档重做日志备份。注意该例子中使用了两个格式子句。第 1 个用于与备份有关的备份集片，第 2 个用于与归档重做日志有关的备份集片。在默认使用 FRA 情况下，上述格式子句是需要的，因为如果使用 keep 参数，则归档重做日志备份备份集片和数据库备份备份集片不能存在于 FRA 中。而在 11g 中则不需要：

```
backup database tag='keep_full' format 'c:\oracle\backup\%U'
    keep until time = 'sysdate+180' logs plus archivelog
    format ='c\oracle\backup\%U';
```

无论使用何种版本，您都希望知道某一个备份被设置为保存多久。List backup of database 命令将提供该类信息。在第 18 章中将会更加详细地讨论 list 命令，这里的示例可能是您希望看见的输出结果。注意在返回输出中，有一行指出"Keep: LOGS"，紧接着显示"Until"和日期。表明这是 Oracle Database 11g 的一个归档备份，该备份在以前的 Oracle 10g 或更早的版本中使用 keep logs 选项。该备份将保存到报告中 Until 项指出的时间为止。

```
BP Key: 36 Status: AVAILABLE Compressed: NO Tag: KEEP_FULL
            Piece Name: C:\ORACLE\BACKUP\1FKJVSSA_1_1
```

```
Keep: LOGS                Until: 08-JAN-10
  List of Datafiles in backup set 36
File LV Type Ckp SCN Ckp Time Name
---- -- ---- ---------- --------- ----
1 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\SYSTEM01.DBF
2 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\SYSAUX01.DBF
3 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\UNDOTBS01.DBF
4 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\USERS01.DBF
5 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\EXAMPLE01.DBF
6 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\UNDO_NEW_01.DBF
7 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\SMALL01.DBF
8 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\SMALLTWO.DBF
9 Full 4358741 12-JUL-09 C:\ORACLE\ORADATA\MY\SMALL_THREE01.DBF
```

可以如下例所示查询 V$BACKUP_SET 及 V$BACKUP_PIECE 视图。也可以通过 RC* 视图通过恢复目录模式查询同样的内容：

```
select a.set_stamp, a.set_count, a.backup_type, a.pieces, b.piece#, a.keep_
until,
a.keep_options
from v$backup_set a, v$backup_piece b
where a.set_stamp=b.set_stamp
and a.set_count=b.set_count
and a.keep='YES';
SET_STAMP SET_COUNT B        PIECES     PIECE# KEEP_UNTIL       KEEP_OPTION
---------- ---------- - ---------- ---------- ------------------ -------
691981428         26 D          1          1 01/08/2010 01:03:48 BACKUP_LOGS
691982114         27 D          1          1 01/08/2010 01:15:13 BACKUP_LOGS
691982117         28 L          1          1 01/08/2010 01:15:17 BACKUP_LOGS
691982121         29 D          1          1 01/08/2010 01:15:20 BACKUP_LOGS
691982247         30 D          1          1 01/08/2010 01:17:26 BACKUP_LOGS
691983183         31 D          1          1 01/08/2010 01:33:02 BACKUP_LOGS
691983206         32 L          1          1 01/08/2010 01:33:25 BACKUP_LOGS
691983252         33 D          1          1 01/08/2010 01:34:11 BACKUP_LOGS
692023212         36 D          1          1 10/20/2009 12:40:12 LOGS
692024018         37 D          1          1 10/20/2009 12:40:12 LOGS
692056970         47 D          1          1 01/08/2010 22:02:50 LOGS
692057685         48 D          1          1 01/08/2010 22:02:50 LOGS
```

11.5.8 归档日志删除策略

从 Oracle Database 11g Release 1 开始，不仅仅只能在 Oracle 备用数据库(第 21 章有关于 RMAN 与 Oracle 备用数据库的更多的内容)上配置归档重做日志的保留策略。从 Oracle Database 11g Release 1 开始，可以配置应用到一般的归档重做日志的保留策略。

可以使用 configure 命令配置归档日志删除策略。保留策略对所有归档重做日志都有效，包括在 FRA 中的归档重做日志。RMAN 将自动删除 FRA 中的归档重做日志，如果未使用 FRA，则需要使用 delete obsolete 命令从其他目录中手动删除过时的归档重做

日志。

归档重做日志删除策略要基于 RMAN 备份归档重做日志的次数来确定。默认情况下，归档日志删除策略值设置为 none 值，设置为该值意味着需要手动执行对归档重做日志的删除操作(例如，在备份期间，使用 RMAN 的 delete input 命令)。如果希望确保至少存在两个归档重做日志的备份后将其标记为 obsolete，则可以执行以下 configure 命令：

```
Configure archivelog deletion policy to backed up 2 times to device type
  disk;
```

注意受到备用数据库有关的保留策略影响的日志不会影响保留策略。也需要注意当归档重做日志被标记为 obsolete 时，Oracle 不会立即将其删除，在 FRA 中的空间被耗尽时，过时的归档重做日志才会被删除。也就是说归档重做日志将尽可能地被保存在 FRA 中。

11.5.9 重写 configure exclude 命令

通过执行 configure exclude 命令(第 3 章讨论过这个命令)，可以配置 RMAN，排除上一次备份以来没有发生变化的数据文件。如果要确保 RMAN 备份这些数据文件，可以在下面的 backup 命令中添加 noexclude 参数：

```
backup database noexclude keep forever tag='test backup';
```

11.5.10 使用 backup 命令检查数据库的错误

另一个便利的 RMAN 功能是不需要执行备份操作就可以使用 RMAN 来扫描数据库的物理错误和逻辑错误。使用 backup 命令的 validate 参数以及 check logical 选项可以实现这个功能。下面是一个使用该选项的示例：

```
backup validate check logical database;
```

注意：
在 RMAN validate 操作期间会生成一些文本，看起来像是创建了一个备份集，但事实并非如此。在 RMAN validate 操作期间不会生成任何备份文件片。

11.5.11 跳过脱机的、不可访问的或只读的数据文件

有些时候，数据库中的数据文件可能处于 ONLINE 以外的状态。如果数据文件是只读的，在备份数据库时可能就不需要每次都备份这些只读的数据文件。如果数据文件是脱机的或不可访问的，假如没有执行特定的操作来指示 RMAN 跳过这些缺少的数据文件，RMAN 备份操作就会失败。因此，需要使用 skip 参数，使用该参数可以根据需要跳过脱机的、只读的或不可访问的数据文件(或者所有这些数据文件)。下面是使用 skip 参数完成这一功能的一些示例。

```
backup database skip readonly;
```

```
backup database skip offline;
backup database skip inaccessible;
backup database skip readonly skip offline skip inaccessible;
```

inaccessible 参数可以使 Oracle 跳过不能读取的数据文件，这些文件没有实际地存在于磁盘上(例如数据文件已经从磁盘上删除了，或者已经将数据文件移到另一个位置)；使用 offline 参数可以跳过脱机的但是仍然物理存在的数据文件；skip readonly 参数可以使 Oracle 跳过备份只读的数据文件。当然，可以使用 configure 命令来指示 Oracle 不要备份只读的表空间，下一小节将介绍如何备份只读的数据文件。

注意：
在先前的 Oracle 数据库版本中，若迁移表空间处于只读状态，则 RMAN 不能对其进行备份。从 Oracle Database 11g 开始，这一情况已经不再是问题了。可以使用 RMAN 备份迁移表空间，即使该表空间处于只读模式。

11.5.12　强制备份只读的数据文件

在上一小节中，我们介绍了如何使备份操作跳过只读的数据文件，但是这样有些乏味。为此，Oracle 提供了更方便的备份优化。在第 3 章介绍 configure 命令时提到过备份优化，备份优化会使 RMAN 默认不备份没有发生变化的表空间(例如只读表空间)。但如果要执行一个忽略这个配置的特定备份操作，可以使用 force 参数来确保备份所有数据文件。下面是一个示例：

```
backup database force;
```

11.5.13　基于上次备份时间来备份数据文件

如果愿意的话，Oracle 会允许用户在备份进程中指示只备份给定时间以来没有被备份过的数据文件。如果在数据库中添加了新的数据文件(这一部分首先讨论这一点)，或者只想备份给定天数内发生变化的数据文件，使用 Oracle 的这个功能都是非常方便的。下面将详细介绍这些问题。

1. 只备份添加的新数据文件

我们可以使用一个简洁的选项。假设在数据库中添加了 4 个或 5 个新数据文件，我们希望不要备份整个数据库，而是备份这些新数据文件。当然，可以只备份单个的数据文件(本章后面将介绍这个问题)，不过还有一种更简单的方法：使用 backup 命令的 not backed up 选项，这样 RMAN 就只备份没有被备份过的数据文件。下面是一个示例：

```
backup database not backed up;
```

2. 备份指定时间周期内没有被备份的数据文件

另外，我们可能需要在指定的时间(例如晚上)备份指定的数据文件。如果需要重新

启动一个失败的备份操作，使用 since time 选项也是非常方便的。如果备份操作失败，则可以在排除故障后使用 since time 选项来重新启动这个备份操作。例如，假设您的磁带系统在两天前的备份操作中死机，您就可以在修复磁带系统后执行下面的命令来重新启动这个备份操作：

```
backup database not backed up since time='sysdate - 2';
```

这时，RMAN 会备份最近两天内没有备份过的那些数据文件。需要注意的是，我们可以使用数据库 NLS_DATE 格式来表示时间，也可以使用 SQL 日期表达式(如示例所示)来表示时间。在 since time 选项中，还可以使用另外一个参数来备份归档日志，以确保每个归档日志在被删除前已备份了一定的次数，本章后面将介绍这个问题。

一些备份策略包括在创建归档的重做日志后立刻对其进行备份，也可能在每夜的备份之后备份这些归档的重做日志。一旦在每晚的备份之后备份这些归档的重做日志，它们就将被删除。默认的配置保留策略实际上不允许这样配置数据库，但是可以在 backup 命令中手动地定义保存标准。作为示例，我们可以使用 backup archivelog 和 backup database 命令执行这种备份，如下所示：

```
-- This command backs up archived redo logs not already backed up.
Backup archivelog all not backed up 1 times;
-- Later in the day, back up the database and archived redo logs, and
-- remove the archived redo logs
backup as compressed backupset database plus archivelog
not backed up 1 times delete input;
```

3. 在备份操作期间检查逻辑损坏

在默认情况下，RMAN 会检查数据库数据块的逻辑损坏。如果发现了任何损坏，备份操作就会失败，这是默认情况。如果需要更多的错误检查，则可以使用 backup 命令的 check logical 选项来配置备份操作执行逻辑损坏检查。下面是使用 check logical 选项的两个示例：

```
backup check logical database;
backup validate check logical database;
```

注意：

若以备用数据库配置方式运行数据库，则 Oracle Database 11g Release 2 对损坏块提供自动修复功能。若能确认损坏块在备用数据库上或是在主数据库上，将使用未受影响的数据块修复。关于该新特性可参考第 15 章以获取更多的信息。

在第一个示例中，RMAN 会在检查逻辑损坏的同时物理备份数据库。在第二个示例中，RMAN 只验证数据库数据块，它只执行逻辑数据库验证操作，而不执行实际的数据库物理备份操作。需要注意的是，如果要在给定的错误数内仍然继续执行备份操作，就需要先设置 maxcorrupt 参数值。如下所示，需要使用 run 代码块来设置 maxcorrupt 参

数值：

```
run {
    set maxcorrupt for datafile 1,2,3,4,5,6,7 to 10;
    backup validate check logical database;
    }
```

11.5.14　在 RMAN 复制设备上生成备份副本

有时，我们可能需要创建备份集片的多个副本。这可以通过配置默认地生成，也可以使用 copies 参数对某个备份进行配置来创建备份片的多个副本(还可以使用 set backup copies 参数来完成这个功能)。下面是一个使用 copies 选项生成备份副本的示例：

```
backup database copies=2;
```

11.5.15　捕获隐蔽的控制文件

include current controlfile 选项会创建当前控制文件的快照，并且会将这个控制文件放入使用 backup 命令生成的每个备份集中。下面是一个使用该命令的示例：

```
backup database device type disk include current controlfile;
```

在默认情况下，如果备份数据文件 1，无论如何都会备份控制文件。因此，如果要执行表空间备份操作或数据文件备份操作，使用 include current controlfile 参数就非常方便。此外，如果配置了控制文件的自动备份操作，上面的命令还会将当前控制文件存储在备份集中(这样我们就拥有控制文件的两个副本，尽管在 ARCHIVELOG 模式中这两个副本可能稍有不同)。

11.6　set 命令介绍

我们已经介绍了脱机备份操作的许多内容。至此，对有关在备份时 RMAN 命令的工作原理已有了一些了解。在讨论联机备份操作前，还要简单介绍一下 RMAN 的 set 命令。

使用 set 命令可以定义只应用于当前 RMAN 会话的设置。换句话说，set 命令与 configure 命令(参阅第 3 章)非常相似，但是 set 命令的设置不是永久的。根据使用的 set 命令的实际需要，有两种使用 set 命令的方法。可以使用 set 命令在 run 代码块外，执行下面这些操作：

- 使用 set echo 命令在消息日志中显示 RMAN 命令。
- 使用 set dbid 命令指定一个数据库的数据库标识符(database identifier，DBID)。

某些 set 命令只能在 run 代码块的限定范围内使用。最常见的 set 命令有：

- **set newname 命令**：用于执行表空间时间点恢复(TSPITR)或数据库复制操作。该命令允许指定新的数据库数据文件名。将数据库移动到新的系统中并且文件

系统名不同时，我们可以使用 set newname 命令。使用 set newname 命令时还需要使用 switch 命令，本章后面部分将用示例来说明这个问题。

- **set maxcorrupt for datafile 命令**：使用该命令可以定义 RMAN 操作失败前所允许的数据块损坏的最大数目。
- **set archivelog destination 命令**：使用该命令可以修改归档重做日志的 archive_log_dest_1 目标。
- **set 命令和 until 子句**：使用 set 和 set 命令的 until 子句可以定义数据库时间点恢复操作期间所使用的具体时间点、SCN 或日志序列号。
- **set backup copies 命令**：使用该命令可以定义为备份集中的每个备份片应当创建的备份文件的副本数。
- **set command id 命令**：使用该命令可以关联给定的服务器会话和给定的信道。
- **set controlfile autobackup format for device type 命令**：使用该命令可以修改用于控制文件自动备份操作的默认格式。

执行备份操作时，可能会使用上面的这些命令。例如，如果要执行一个为每个备份片创建两个副本的备份操作，并且允许数据文件 3 的最大损坏数为 10，就可以创建如下所示的备份脚本：

```
run
{
set maxcorrupt for datafile 3 to 10;
set backup copies = 2;
backup database;
}
```

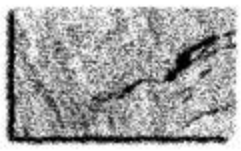
11.7　联机 RMAN 数据库备份

本章的前半部分介绍了脱机备份操作和 set 命令，本节(和下一节)将介绍联机备份操作。在学习联机备份操作以前，有必要了解前面所讲述的脱机备份操作的知识。如果是直接浏览本章的这一部分，就需要返回并阅读前面的内容。如果已经学习了本章的前半部分并且需要时间来掌握这些内容，则可以在继续学习之前先休息一下。

本节首先讨论几种不同的联机备份操作：数据库的整体备份、表空间备份、数据文件备份；然后介绍归档日志文件备份以及控制文件和参数文件的备份。

11.7.1　联机数据库备份

第 1 章和第 3 章曾经详细介绍过，要使用 RMAN 执行联机备份操作，数据库就必须置于 ARCHIVELOG 模式中。如果数据库不在 ARCHIVELOG 模式中，则尝试联机备份时 RMAN 就会生成一个错误。因此，在开始执行 RMAN 联机备份时，确保数据库被置于 ARCHIVELOG 模式。



注意：
在本章中从这一小节开始，只要没有特别说明，就假定已经配置了默认的信道(参阅第 3 章)，这样就不必再输入如 allocate channel 之类的命令，以更多地介绍其他重要信息。

可以发现：联机备份操作并不是与脱机备份操作完全不同。实际上，联机备份操作稍微简单一些，这是因为我们不必关闭数据库，再加载数据库。如下所示，只要配置了默认值(参阅第 3 章)，就可以非常简单地执行一个联机备份操作：

```
backup database plus archivelog;
```

上面的这条命令完成了所有的操作。首先，进程使用 alter system archivelog current 命令完成了日志切换操作；然后，进程将现存的所有归档重做日志备份；接下来，开始执行实际的数据库备份操作。此时会发生另一个日志切换操作(使用 alter system archivelog current 命令)，同时 RMAN 备份剩余的归档重做日志(使用 backup archivelog all 命令)；最后，执行控制文件和服务器参数文件的自动备份操作。由于一个完整的数据库备份通常含有数据文件 1(数据文件 1 属于 SYSTEM 表空间)，所以完全数据库备份中通常会存在一个控制文件和一个服务器参数文件的备份。

RMAN 实习：执行一个联机备份操作

实习注意事项

本实习假定已经为数据库配置了第 3 章所介绍过的自动信道，且为备份操作配置了一个名为 backup_admin 的数据库账户(第 3 章也曾做过介绍)，此外还假定使用已配置完的 MML 层。最后，数据库必须配置为在 ARCHIVELOG 模式中操作。

步骤 1　启动 RMAN:

```
C:\>rman target=backup_admin/robert
```

步骤 2　启动备份操作:

```
RMAN> backup database plus archivelog;
```

下面是一个按照这些步骤所执行的联机 RMAN 备份的完整示例:

```
C:\>rman target=backup_admin/robert
RMAN> backup database plus archivelog;
Starting backup at 15-JUN-02
current log archived
using target database controlfile instead of recovery catalog
configuration for DISK channel 3 is ignored
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=13 devtype=DISK
channel ORA_DISK_1: starting archive log backupset
channel ORA_DISK_1: specifying archive log(s) in backup set
```

```
input archive log thread=1 sequence=351 recid=13 stamp=464457020
channel ORA_DISK_1: starting piece 1 at 15-JUN-02
input archive log thread=1 sequence=352 recid=14 stamp=464609012
input archive log thread=1 sequence=353 recid=15 stamp=464609115
channel ORA_DISK_1: finished piece 1 at 15-JUN-02
piece handle=D:\BACKUP\ROBT\BACKUP_20DR2QJ8_1_1 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:11
channel ORA_DISK_1: starting archive log backupset
channel ORA_DISK_1: specifying archive log(s) in backup set
input archive log thread=1 sequence=357 recid=19 stamp=464610450
input archive log thread=1 sequence=358 recid=20 stamp=464611007
input archive log thread=1 sequence=359 recid=21 stamp=464611921
channel ORA_DISK_1: starting piece 1 at 15-JUN-02
channel ORA_DISK_1: finished piece 1 at 15-JUN-02
piece handle=D:\BACKUP\ROBT\BACKUP_22DR2QJK_1_1 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 15-JUN-02
Starting backup at 15-JUN-02
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
input datafile fno=00001 name=D:\ORACLE\ORADATA\ROBT\SYSTEM01.DBF
input datafile fno=00005 name=D:\ORACLE\ORADATA\ROBT\USERS01.DBF
input datafile fno=00004 name=D:\ORACLE\ORADATA\ROBT\TOOLS01.DBF
input datafile fno=00003 name=D:\ORACLE\ORADATA\ROBT\INDX01.DBF
input datafile fno=00002
name=D:\ORACLE\ORADATA\ROBT\ROBT_TEST_RECOVER_02.DBF
input datafile fno=00010 name=D:\ORACLE\ORADATA\ROBT\ROBT_RBS_01.DBF
input datafile fno=00011
name=D:\ORACLE\ORADATA\ROBT\ROBT_TEST_TBS_01.DBF
input datafile fno=00007
name=D:\ORACLE\ORADATA\ROBT\ROBT_TEST_RECOVER_01.DBF
input datafile fno=00006
name=D:\ORACLE\ORADATA\ROBT\ROBT_TEST_RECOVER_03.DBF
channel ORA_DISK_1: starting piece 1 at 15-JUN-02
channel ORA_DISK_1: finished piece 1 at 15-JUN-02
piece handle=D:\BACKUP\ROBT\BACKUP_23DR2QJU_1_1 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:04:56
Finished backup at 15-JUN-02
Starting backup at 15-JUN-02
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archive log backupset
channel ORA_DISK_1: specifying archive log(s) in backup set
input archive log thread=1 sequence=360 recid=22 stamp=464612416
channel ORA_DISK_1: starting piece 1 at 15-JUN-02
channel ORA_DISK_1: finished piece 1 at 15-JUN-02
piece handle=D:\BACKUP\ROBT\BACKUP_25DR2R2K_1_1 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:04
Finished backup at 15-JUN-02
```

```
Starting Control File and spfile Autobackup at 15-JUN-02
piece handle=D:\BACKUP\ROBT_C-3395799962-20020615-02 comment=NONE
Finished Control File and spfile Autobackup at 15-JUN-02
```

上面介绍了一个完整的联机数据库备份操作，下一小节将介绍表空间备份。

注意：
完全数据库备份与增量备份(本章后面将作介绍)存在不同是因为不能将完全数据库备份作为增量备份的基础。

11.7.2 表空间备份

某些情况下，希望执行表空间备份操作，而不是数据库的整体备份。在表空间中释放一个分区前，或者在表空间变为只读属性后，就可能需要执行表空间备份操作。如下所示，我们可以简单地使用具有 tablespace 参数的 backup 命令来执行一个表空间备份操作：

```
backup tablespace users;
```

如果同时要备份归档重做日志，可以执行下面的命令：

```
backup tablespace users plus archivelog;
```

如下所示，我们要确保备份的是当前控制文件：

```
backup tablespace users include current controlfile plus archivelog;
```

当然，我们不是要真正地备份一个表空间，而是要备份与这个表空间关联的数据文件。Oracle 只是将表空间名转换为一个与表空间关联的数据文件的列表。通常，如果没有配置自动备份控制文件(参阅第 3 章)并且不备份数据文件 1，就不会执行控制文件备份操作。当然，也可以在 backup 命令中使用 include current controlfile 参数来备份控制文件。

注意：
RMAN 不会禁止在 NOARCHIVELOG 模式中备份表空间或数据文件(只要没有打开数据库)。不过，当数据库置于 NOARCHIVELOG 模式时，这些备份实际上用途不大(除非同时备份了所有的表空间和数据文件)。

11.7.3 数据文件备份

有时，用户可能希望备份数据库中具体的数据库数据文件。或者我们可能要将数据库数据文件移动到新的设备中，因此要在移动前备份这些数据文件。RMAN 允许使用带有 datafile 参数的 backup 命令来备份数据文件，datafile 参数后面是要备份的文件名或数据文件号。下面是一些数据文件的 backup datafile 命令示例：

```
backup datafile 2;
backup datafile 'd:\oracle\oradata\robt\users01.dbf';
backup datafile 'd:\oracle\oradata\robt\users01.dbf' plus archivelog;
```

同样，如果备份数据文件 1，或者配置了自动备份控制文件，就会在备份数据文件的同时备份控制文件和服务器参数文件。在上面的最后一个示例中，同时还会备份归档的重做日志。

11.7.4 归档重做日志备份

由于各种原因，我们可能需要备份归档的重做日志，但不备份数据库，这时可以使用 backup archivelog 命令。要备份所有归档的重做日志，可以简单地执行 backup archivelog all 命令。我们还可以备份指定范围内的归档的重做日志，这里有多个选项，这些选项包括时间、SCN 或重做日志序列号(或者表示其范围的值)。特定的选项为 from SCN 选项，from sequence 选项和 from time 选项。记住使用 from 选项可以使一些归档重做日志留在磁盘中而不被备份。下面是一些备份归档的重做日志的示例：

```
backup archivelog all;
backup archivelog from time 'sysdate - 1';
backup archivelog from sequence 353;
```


注意：
Oracle Database 11*g* Release 2 提供的新特性之一是如果 FRA 中的归档重做日志损坏，RMAN 将“故障切换”到其他定义的归档重做日志目标上。如果该归档重做日志在目标目录中且没有损坏，则 RMAN 将从该源目标目录中备份归档重做日志。

一旦备份了归档的重做日志，就可能会使用 RMAN 删除这些归档的重做日志。delete input 选项可以完成这一操作。此外，该选项也可以用于删除数据文件副本或备份集副本(本章后面部分将介绍副本的有关内容)。下面是两个在归档的重做日志备份命令中使用 delete input 参数的示例：

```
backup archivelog all delete input;
backup archivelog from sequence 353 delete input;
```

也可以指示 RMAN 备份归档重做日志的多个副本。在下面的示例中，我们使用 backup 命令的 not backed up *n* times 参数的来确保至少备份 3 次重做日志，并且不会再次备份已经备份过 3 次的归档的重做日志：

```
backup archivelog not backed up 3 times;
```

此外，还可以使用 backup 命令与 until time 参数来确保归档的重做日志在磁盘上保留一定的天数，如下所示：

```
backup archivelog until time 'sysdate - 2'delete all input;
```

注意：
not backed up 参数与 delete input 参数相互排斥。不管归档的重做日志的备份次数为多少，delete input 参数都会删除这个归档的重做日志。

11.7.5　控制文件和服务器参数文件备份

与归档的重做日志相似，有时候可能只希望备份控制文件或服务器参数文件，RMAN 同样也提供了用于这一功能的特殊命令。使用 backup spfile 命令可以备份服务器参数文件。例如，如果修改了数据库的配置，使用该命令执行备份操作就非常方便。要备份控制文件，需要使用 backup 命令的 current controlfile 参数来生成当前控制文件的一个副本。current controlfile 参数还提供了一个 for standby 子句，这个子句会创建一个备用数据库要用到的备份控制文件。

可以使用 backup 命令的 controlfilecopy 参数来创建一个含有外部创建的控制文件备份的备份集，执行 alter database backup controlfile to 'file_name' SQL 命令或 RMAN 的 copy 命令(本章后面部分将做介绍)可以创建这个控制文件备份。此外，还可以备份由 alter database create standby controlfile 命令创建的备用数据库控制文件，该功能的优点在于可以使用外部控制文件备份文件，在 RMAN 中注册这些文件，然后创建一个含有控制文件备份的备份集。下面是使用 controlfilecopy 参数的一些示例：

```
backup current controlfile;
    sql "alter database backup controlfile
    to ''d:\backup\robt\contf_back.ctl''";
    backup controlfilecopy 'd:\backup\robt\contf_back.ctl';
```

11.7.6　备份集备份

我们可能首先在磁盘上备份数据，然后在磁带上备份备份集。RMAN 通过使用 backup 命令来支持上述的操作。例如，假设执行一条 backup database 命令，由于配置的默认设备是磁盘，所以会在磁盘上存储整个备份集。现在我们要将备份集移动到磁带上，为此需要使用带有 backupset 参数的 backup 命令，执行这条命令后，Oracle 会在为备份操作所分配的信道上备份所有备份集。

使用 backup backupset 命令可以选择备份所有备份集，也可以选择备份特定的备份集。另外，只能将备份集从磁盘备份到磁盘，或者将备份集从磁盘备份到磁带，但不支持磁带到磁带或磁带到磁盘的备份操作。在前面介绍归档日志备份时所讨论的 delete input 选项也适用于备份集备份。在备份集备份中，使用 delete input 选项可以在备份操作成功后删除源备份集的文件。下面是使用 backup backupset 命令的一些示例：

```
backup backupset all;
    backup backupset all
    format='d:\backup\newbackups\backup_%U.bak'
    tag='Backup of backupsets on 6/15'  channel 'ORA_DISK_1';
    backup backupset completed before 'sysdate - 2';
    backup backupset completed before 'sysdate - 2' delete input;
    backup backupset completed after 'sysdate - 2' delete input;
```

这里的一个备份策略示例可能是在磁盘上执行 RMAN 备份操作，然后再使用 backup

backupset 命令将备份集备份到磁带上。我们可能要将备份集在磁盘上保存两天，然后执行下面两条命令：第一条命令 backup backupset completed before 'sysdate-2'会备份最近两天的备份集；第二条命令 backup backupset completed after 'sysdate-2' delete input 则会备份和删除两天前的所有备份集，这个命令生成旧备份集的一个最终备份，然后在磁盘上删除这些旧备份集。

注意：
如果要在其他地方备份控制文件的自动备份，并且要求 RMAN 目录仍然跟踪备份集的位置，使用备份集备份就非常方便。

11.7.7　闪回恢复区备份

RMAN 提供了通过 backup recovery area 命令将整个 FRA 备份到磁带的功能。但并不是 FRA 中的所有文件都备份，备份的文件包括完整备份集和增量备份集、控制文件自动备份、归档的日志，以及数据文件副本。如果备份的 FRA 包含闪回日志、当前的控制文件以及联机重做日志，则不会备份这些文件。注意，必须将这些文件备份到磁带上。

11.8　副本

介绍完所有这些与备份集和备份片有关的新内容后，您可能会提出这样的疑问："可以只生成这些数据库数据文件的一个副本吗？"答案是肯定的。我们可以使用 RMAN 只生成不同数据库结构的副本，下面将介绍这个问题。首先，我们将概述创建副本以代替备份集的各个方面；然后再介绍如何创建数据文件副本、控制文件副本以及归档的重做日志文件副本。

11.8.1　映像副本介绍

RMAN 可以创建数据库数据文件、归档的重做日志或控制文件的精确副本。RMAN 映像副本是这些文件的副本，区别仅仅在于名称和位置的变化上。我们不需要考虑备份片或其他问题。映像副本只能在磁盘上生成，且不能生成增量副本。生成映像副本时必须加载或打开数据库。生成副本的历史记录保存在数据库控制文件中，所以我们可以跟踪副本生成的时间和驻留的位置。

类似于常规备份，可以建立整个数据库、表空间或数据文件的映像副本(这与早期的 RMAN 版本存在很大区别)。RMAN 复制过程提供一些与普通 RMAN 备份集相同的保护，例如检查损坏的代码块以及(可选的)逻辑损坏。此外，映像副本可以与普通的备份集(例如增量备份和归档的重做日志备份)结合起来，帮助完成数据库恢复。

11.8.2　数据库、表空间和数据文件的映像副本

backup 命令支持数据库映像副本的创建。RMAN 可以通过 backup as copy 命令生成

映像副本，该过程类似于执行备份集。下面是使用 RMAN 建立数据库映像副本的示例：

```
RMAN> backup as copy database;
```

RMAN 将使用经过配置的 FRA 存储备份副本。如果使用 FRA，数据文件映像将存储在称为 datafile 的目录中，如同下面数据文件映像副本的部分示例输出所示：

```
RMAN> backup as copy database;
    Starting backup at 12-NOV-05
    using channel ORA_DISK_1
    using channel ORA_DISK_2
    channel ORA_DISK_1: starting datafile copyinput datafile fno=00001
    name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSTEM01.DBF
    channel ORA_DISK_2: starting datafile copy
    input datafile fno=00003
    name=C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSAUX01.DBF
    output Filename=C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2
    \DATAFILE\O1_MF_SYSAUX_1QFBPPON_.DBF tag=TAG20051112T205403 recid=
    2stamp=574203351
    channel ORA_DISK_2: datafile copy complete, elapsed time: 00:01:55
    channel ORA_DISK_2: starting datafile copy
```

表空间的映像副本操作的方式与此类似，只是使用的是 backup as copy 命令和 tablespace 关键字：

```
backup as copy tablespace users;
```

最后，可以创建数据文件的映像副本：

```
backup as copy datafile 1;
    backup as copy datafile
    'C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSTEM01.DBF';
```

注意：
在数据库加载时可以生成映像副本，如果数据库在 ARCHIVELOG 模式中，数据库必须是打开的。

11.8.3　控制文件副本

也可以使用 backup controlfile 命令创建控制文件副本。备份可以在数据库处于加载或打开状态时进行。一般来说，应该使控制文件能够自动备份并允许 RMAN 在备份完成后自动备份控制文件。

若需要手动创建包含控制文件的 RMAN 备份集，则可参考以下示例备份：

```
backup current controlfile;
```

注意该方法与控制文件自动备份不同。使用该方法时，若从控制文件自动备份中还原，则不能恢复备份控制文件的。

也可以创建控制文件的副本。这些副本与使用 alter database backup controlfile to trace 命令创建的备份控制文件相同，可用于恢复目的。以下是一个使用 RMAN 创建的控制文件副本的示例：

```
backup as copy current controlfile
    format 'c:\controlback\control01.ctl' reuse;
```

注意 backup as copy 命令不会重写已经存在的备份控制文件，除非使用 reuse 关键字，正如上例所示。还应当注意，backup as copy 命令不会创建 RMAN 备份集，除非物理文件是备份控制文件。

若希望创建一个控制文件与创建的备用数据库一起使用，则可以使用 for standby 子句。如下所示创建 RMAN 备份集：

```
backup as copy standby controlfile;
```

与使用常规方式创建备份控制文件一样，在创建控制文件备份时，可以指定特定的文件名/位置。使用该方式，将在该位置创建物理文件，而不会创建 RMAN 备份集：

```
backup as copy current controlfile format 'c:\oracle\controlfilecopy
    .ctl';
```

11.8.4 ARCHIVELOG 映像副本

生成归档的重做日志副本是非常有用的。使用 Oracle 的 LogMiner 产品挖掘(mine)归档的重做日志副本显然比必须先从备份集中析取这个归档的重做日志更为简便。使用 copy 命令的 archivelog 参数可以创建归档的重做日志副本。但是，正如前面所介绍的那样，使用 copy archivelog 命令时必须列出每个归档的重做日志名，而不能指定要复制的归档的重做日志的时间范围。下面是一个生成 ARCHIVELOG 文件副本的示例：

```
backup as copy archivelog all;
```

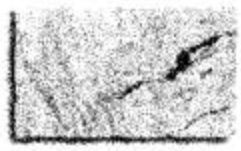
11.9 增量 RMAN 备份

希望通过此书您能够毫不费力地获得数据库的良好备份。现在我们将介绍 RMAN 中的增量备份。通过增量备份，RMAN 允许用户只备份上次增量备份操作以来被修改过的数据块。增量备份具有以下优点：

- 更少地使用磁带或磁盘
- 降低网络带宽要求
- 节约备份时间

既可以联机也可以脱机地在 ARCHIVELOG 模式或 NOARCHIVELOG 模式中执行增量备份操作，这是非常方便的。需要记住的是，一旦选择了增量备份策略，就会存在效益方面的取舍。如果显著减少了整个备份操作的时间(这可能非常重要)，在恢复操作方面就会投入更多的时间，这是因为如果使用了增量备份策略，Oracle 就需要使用多个备

份集来恢复数据库，恢复数据库所需的时间就会显著地增加。



注意：
如果选择执行 NOARCHIVELOG 模式下的数据库增量备份操作，就必须确保每次备份数据库时都按照一致的方式关闭这个数据库。

11.9.1　块更改跟踪文件

默认情况下，当执行增量备份时，发生任何更改的所有数据文件都将被备份。这可能使增量备份花费更长的时间，并且会增加增量备份的大小。RMAN 提供了只备份更改过的数据库块的功能。这就可以加快增量数据库备份的速度并减少其大小。执行 alter database enable block change tracking 命令可启用块更改跟踪。该命令可以创建块更改跟踪文件(BCTF)。

当启用了块更改跟踪文件，可以让 Oracle 命名相关的块更改跟踪文件。Oracle 会使用 OMF 命名标准来命名 BCTF。如果没有使用 OMF，也可以定义块更改跟踪文件的位置和名称，如下所示：

```
alter database enable block change tracking using file
'/u01/app/oracle/block_change/rob10gr2_block_change.fil';
```

如果前面的块更改跟踪文件已经存在，则需要使用 reuse 参数：

```
alter database enable block change tracking using file
'/u01/app/oracle/block_change/rob10gr2_block_change.fil' reuse;
```

使用 alter database disable block change tracking 命令可禁用块更改跟踪。块更改跟踪文件的大小通常预先分配且与数据库的大小和重做日志线程的数量有关。块更改跟踪文件的大小一般是数据库大小的 1/250000。若使用 Oracle Real Application Clusters(RAC)数据库配置，则每个节点需要访问 BCTF。在此配置下，为方便文件共享，一般需要考虑在 ASM 中存储 BCTF。

BCTF 的容量取决于每次 0 级备份之间产生的增量备份的数量。增量备份越多，需要存储在 BCTF 中的与块更改相关的元数据的数量也就越多。

BCTF 可以容纳最多 8 个与备份有关的信息(例如，1 个基本备份和 7 个 1 级增量备份)。当 8 个备份相关信息使用完成后，下一次备份将会删除先前备份的信息，使 BCTF 无效。BCTF 文件将以 10MB 的大小自动增长。不管数据文件的容量多大，BCTF 将会为每个数据文件分配至少 320KB 的存储空间。

Oracle 数据库默认为不使用块更改跟踪文件，可以通过检查 V$BLOCK_CHANGE_TRACKING 视图来确定是否启用了块更改跟踪。STATUS 列指示是否启用块更改跟踪，而 FILENAME 列则包含块更改跟踪文件的文件名。可以通过使用 alter database rename file 命令移动块更改跟踪文件。

```
SQL> select status from v$block_change_tracking;
STATUS
----------
DISABLED
```

11.9.2 基本备份

执行增量备份操作时，首先需要的是增量基本备份(incremental base backup)，今后所有的增量备份都是基于这个基本备份的。每次执行数据库的备份操作时，都可以通过 backup 命令的 incremental 参数来为备份指派一个增量级别标识符。增量备份按级别指派。基本备份的增量级别为 0，并且必须有基本备份才能够执行其他类型的增量备份操作。增量备份的级别通常为 1(有时级别大于 1)。如果没有生成基本备份就尝试执行增量备份操作(这个增量备份操作使用 1 的备份级别)，RMAN 就会自动执行基本备份操作。下面是一个执行基本增量备份操作的示例：

```
backup incremental level = 0 database;
```


注意：
RMAN 之前的版本除了支持 0 级备份和 1 级备份外，还支持其他级别的备份。从 Oracle Database 10g Release 1 开始，除了 0 级备份和 1 级备份外，Oracle 不再支持其他级别的增量备份。

11.9.3 差异增量备份与累积增量备份

现在，我们需要决定如何执行增量备份操作。可以使用下面的两种方法：

- 差异备份
- 累积备份

这是两种执行增量备份操作的不同方法。接下来我们将详细介绍这两种增量备份的类型。

1. 差异备份

差异备份是 RMAN 生成的增量备份的默认类型。对于差异备份来说，RMAN 会备份自上一次同级或低级差异增量备份以来所有发生变化的数据块。理解这个概念可能有一些困难，图 11-1 可以帮助您理解不同备份级别的差异备份操作。

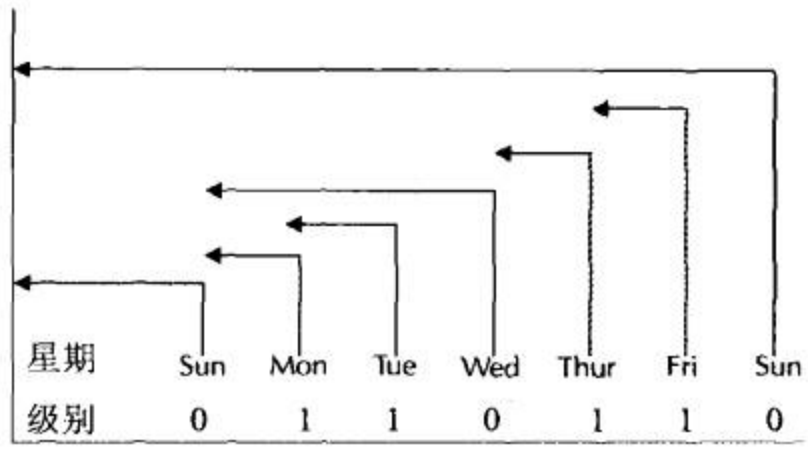


图 11-1　差异备份

在图 11-1 的示例中，我们在星期天执行 0 级差异增量备份操作，这个备份操作会备份整个数据库。根据这个 0 级备份，我们在星期一执行 1 级差异增量备份操作。该备份操作将备份自周日 0 级备份以来所有发生变化的数据块。在周二时 1 级增量份将备份所有自周一 1 级备份以来发生变化的数据块。而在周三将进行另外一次 0 级备份，对整个数据库的所有数据块进行备份。周四和周五将再次执行 1 级备份，仅仅备份发生变化的数据块，与周一和周二所执行的备份操作一致。最后，在周日对所有数据库数据块进行 0 级备份，重新开始新的备份周期。

下面是一个执行 1 级差异增量备份操作的示例，记住如果没有 0 级增量备份，0 级备份将替代 1 级的备份。

```
backup incremental level=1 database;
```

2. 累积备份

RMAN 还提供了另一种增量备份选项：累积备份。使用累积备份可以使备份集备份自上次 0 级备份以来所有发生变化的数据块，忽略之前的 1 级备份。累积备份是一个可选的备份方法，并要求在 backup 命令中使用 cumulative 关键字。同样，理解这个概念也会有一些困难，因此查看另一个示例。图 11-2 可以帮助理解不同备份级别的累积备份操作。

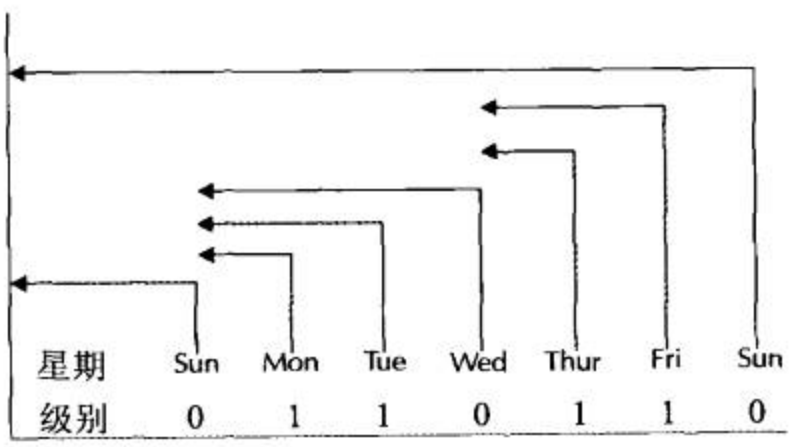


图 11-2　累积备份

与图 11-1 的示例相同，在图 11-2 所示的示例中，我们在星期天执行 0 级差异增量备份操作，此时会备份整个数据库。随后，我们在星期一执行 1 级备份操作，这种备份

不同于差异备份，在星期二进行另一个 1 级差异备份操作。需要注意的是：星期二生成的备份不仅包含星期一的差异备份以后发生变化的数据块，而且还含有星期一生成的备份中所包含的数据块。因此，累积备份累积了任何同级或低级增量备份操作包含的所有发生变化的数据块。如果要执行恢复操作，就只需要星期二生成的备份以及星期天生成的基本备份。我们将继续在这个星期的剩余天数中执行 0 级和 1 级累积备份操作以完成备份策略。

下面是一个执行 1 级累积增量备份操作示例：

```
backup incremental level=1 cumulative database;
```

3. 增量备份选项

Oracle 不仅允许执行数据库的增量备份操作，而且还允许执行表空间、数据文件以及数据文件副本的增量备份操作。控制文件、归档的重做日志以及备份集都不能生成增量备份。此外，可以在执行增量备份操作的同时备份归档的重做日志。下面是一些示例：

```
backup incremental level=0 tablespace users;
backup incremental level=1 tablespace users;
backup incremental level=0 datafile 4;
backup incremental level=1 datafile 4;
backup incremental level=1 database plus archivelog;
```

4. 增量更新备份

RMAN 提供了增量更新备份(也称合并增量备份)。这种备份避免了采用数据文件的完整映像副本进行备份的开销，并且具有与映像副本备份相同的恢复特性。合并的增量备份在默认情况下为累积增量备份。在 Oracle 早期的版本中，如果试图将其视为差异增量备份，将会出现错误。利用合并增量备份，0 级备份(完整备份)，后续创建的备份即 1 级增量备份，当创建这些增量备份后，将会被合并到先前的 0 级备份中。这样就不需要再次创建 0 级备份，从而节省了备份时间。使用块更改跟踪文件，结合合并增量备份可以进一步减少备份数据库所需的时间开销。

在介绍增量更新备份之前，首先看如下的备份示例：

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

recover copy of database 命令没有真正地恢复数据库，但它使 RMAN 将任何增量备份应用于与列出的标记(incr_update)关联的数据文件副本。

先前的命令将会分三个阶段创建备份：

(1) 使用这些命令的首次备份将会创建 0 级备份(假设还没有 0 级增量备份存在时)。注意在备份期间将会出现一些错误，错误以“数据文件没有副本可恢复”开头。这是假设的，因为此时尚无 1 级增量备份存在。

(2) 再次运行上述命令时，将会创建一个 1 级增量备份，除此以外将不会有其他备份产生。当然，仍然会出现第(1)步出现的错误。

(3) 在第(3)步及随后的备份的迭代步骤中，先前的 1 级增量备份将会应用到 0 级备份中。结果是 0 级备份将会按照已应用的 1 级增量备份的结果更新。将产生新的 1 级增量备份。这意味着在执行恢复/还原操作时仅需要 0 级备份和 1 个 1 级增量备份(以及必需的归档重做日志)就可以实现。这种方式显著地减少了还原数据库所需的时间。

注意使用标记标识这些备份。需要明确的是在 recover 和 backup 命令中使用相同名称的标记。

也可以修改命令的顺序，首先执行 backup incremental 命令，然后执行 recover copy of database 命令。如下所示：

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

以此方式修改命令的顺序将会导致 1 级增量备份立即被应用到 0 级基本备份中，将增量备份应用到基本备份将不会有任何延迟。这种方式使得基本备份尽可能与当前数据库一致。

注意该策略假定保留策略为 1 级冗余。如果需要更复杂的保留策略，可以使用 until 子句确保能够满足恢复窗口的要求。例如，假设保留策略是 7 天，可以将保留策略设置为 1 级冗余。如下例所示，使用 until 子句对脚本进行调整：

```
RUN {
    BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG 'incr_update'
    DATABASE;
    RECOVER COPY OF DATABASE WITH TAG 'incr_update' until time "sysdate-8";
    }
```

Metalink note 7457989.1 提供了关于保留策略的更详尽的讨论。若保留策略基于时间，则需要注意仍然要将冗余设置为 1 级，然后使用 until 子句确保能够满足恢复窗口的要求。使用恢复窗口或者大于 1 级的冗余将会导致文件不会从 FRA 中删除。结果空间很快就会被耗尽。

Metalink note 7457989.1 也提供了一些关于块更改跟踪文件的信息，以及使用跟踪文件将会带来的限制。默认情况下，Oracle 的块更改跟踪文件只能跟踪 recover 命令之间不超过 8 天的位图变化情况，如果需要恢复窗口大于 7 天，或者需要执行额外的 1 级增量备份，则需要对参数 bct_bitmaps_per_file 进行修改以允许跟踪更多的位图变化情况。

RMAN 实习：执行一个增量备份操作

实习注意事项

本实习假定已为数据库配置了第 3 章所介绍的自动信道，且为备份操作配置了一个

名为 backup_admin 的数据库账户(第 3 章也曾做过介绍),此外还假定使用配置好的 MML 层。最后，数据库必须配置为在 ARCHIVELOG 模式中运行。

步骤 1 启动 RMAN:

```
C:\>rman target=backup_admin/robert
```

步骤 2 如下所示，执行级别为 0 的增量备份操作，在备份集中包含归档的重做日志，然后在增量备份操作后删除这些归档的重做日志:

```
backup incremental level = 0 database plus archivelog delete input;
```

步骤 3 第二天执行级别为 1 的增量备份操作。如下所示，再次在备份集中包含归档的重做日志，然后在增量备份操作后删除这些归档的重做日志:

```
backup incremental level=1 database plus archivelog delete input;
```

上面简要介绍了 RMAN 备份。第 10 章将介绍如何还原这些备份以及恢复数据库。

11.10 备份操作练习

我们已经介绍了许多不同的备份选项，您可能想知道应从哪里开始着手。我们建议您使用测试数据库来开始练习备份操作，这个测试数据库应该是一个不重要的数据库，最好是专门为练习而创建的。下面介绍的 RMAN 实习将介绍如何执行备份操作。

RMAN 实习：数据库备份操作练习

实习注意事项

这个实习列出了执行数据库备份操作需要实现的所有功能。我们在前面的章节中已经陆续介绍了这些内容，这里将概述所有重要的内容。

步骤 1 在测试数据库中建立一个特殊的账户，这个账户负责执行备份操作。按照第 3 章的说明配置这个账户。需要注意的是，虽然可以选择使用 SYS 账户，但应使用一个独立的账户。

步骤 2 建立 MML 层，并确定执行备份和恢复命令时需要完成的功能。

步骤 3 如第 3 章所述,使用 configure 命令为每一个备份数据的设备(除了支持多信道的特定设备)配置一个单独的默认信道。如第 3 章所述，设置默认并行级别，这样就会分配多个信道来并行执行备份操作。根据需要可以配置备份集的大小。

步骤 4 使用 configure 命令配置自动备份控制文件和服务器参数文件(如果正在使用一个服务器参数文件)。我们暂时先在默认位置创建这些备份，以后会修改存储这些备份的位置。

步骤 5 确认操作系统备份可以备份 Oracle RDBMS 软件(同时也会保证备份控制文件备份)。

步骤 6　建议您最初不要使用恢复目录。一旦正确设置了 RMAN 基本元素，并准备在企业中部署 RMAN，就需要考虑使用恢复目录。

步骤 7　在前几次练习中，如果可以的话，在 NOARCHIVELOG 模式中运行数据库。关闭和加载数据库，并且执行一些 backup database 命令。确认在指定的位置创建备份集。

步骤 8　参阅第 13 章的内容并恢复数据库。建议您首先使用 cp 或 copy 命令来执行整个数据库、控制文件以及联机重做日志的冷备份。

步骤 9　一旦熟练掌握了恢复 NOARCHIVELOG 模式数据库的方法，您就可以将数据库置于 ARCHIVELOG 模式并且再次执行相同的操作。使用 backup database plus archivelog 命令备份这个数据库。

步骤 10　参阅第 13 章和第 15 章的内容执行一些恢复操作，熟练掌握 RMAN 的恢复操作。

步骤 11　练习使用 crosscheck 命令、list 命令以及 report 命令(第 17 章和第 18 章详细介绍了这些命令的用法)，掌握这些命令的用法与用途。

步骤 12　如果您面对的是一个大型企业环境(比如说超过 10 个数据库的环境)，就需要在这个环境中添加和连接恢复目录，并使用这个恢复目录来备份数据库。除了要求依赖于 RMAN 版本使用恢复目录模式之外，强烈建议将一个独立的恢复目录用于开发/测试和产品数据库。同样，再次建议您在正式执行备份操作前熟悉使用恢复目录的备份和还原操作。

步骤 13　一旦熟悉了 RMAN 的使用方法，就可以创建自动执行和调度备份的脚本。例如，如果使用的是 Unix 操作系统，通过 cron 可以调度如下所示的脚本(这里将介绍脱机脚本和联机脚本)：

```
# For offline backups, use this script
#!/bin/ksh
# for offline backups, avoid shutdown aborts at all costs!
rman target rman_backup/password<<EOF
shutdown immediate
startup mount
backup database;
alter database open;
quit
EOF
```

如果执行的是联机备份，就使用下面的脚本：

```
#!/bin/ksh
rman target rman_backup/password<<EOF
backup database plus archivelog;
quit
EOF
```

上面介绍的是 korn shell 脚本，它是 Unix 中一种常用的 shell 脚本编程语言。当然，RMAN 可以在不同的平台上运行，所以我们可以使用自己熟悉的外壳脚本语言。korn 外

壳脚本使用一个称为 here 的文档，该文档含有 EOF(第一个脚本的第四行和第二个脚本的第二行都使用了 EOF)。使用 here 文档好比用户正在这里进行输入，从 EOF 开始的首行到 EOF 结束行之间的内容，即用户的输入。如下所示，您也可以选择只使用文本编辑器所创建的命令脚本，并从 RMAN 命令行调用这个命令脚本:

```
rman target rman_backup/password @backup.cmd
```

下面是用于脱机备份操作的 backup.cmd 脚本:

```
shutdown immediate
startup mount
backup database;
alter database open;
quit
```

对于热备份操作来说，backup.cmd 脚本应当如下所示:

```
backup database plus archivelog;
quit
```

使用 RMAN 的 create script 命令也可以在 RMAN 中存储脚本。由于在 RMAN 中存储脚本需要使用恢复目录，所以第 11 章将讨论在 RMAN 中存储的脚本。

步骤 14 将备份移动到产品数据库之前，应对在测试环境中创建的备份进行还原测试，确认一切操作正常。建议在成功完成还原测试前同时使用旧的备份策略。

步骤 15 将 RMAN 备份策略小心地移动到产品数据库中，每次只能在一个操作系统上执行该操作。首先选择使用“可视程度”最低的系统。建议您在转向使用产品数据库之前先执行多次恢复操作的测试，并不停地双重备份每个数据库直到能用 RMAN 备份在测试中成功地恢复这个产品数据库。还有，如果需要，可考虑执行归档的重做日志的单独备份。

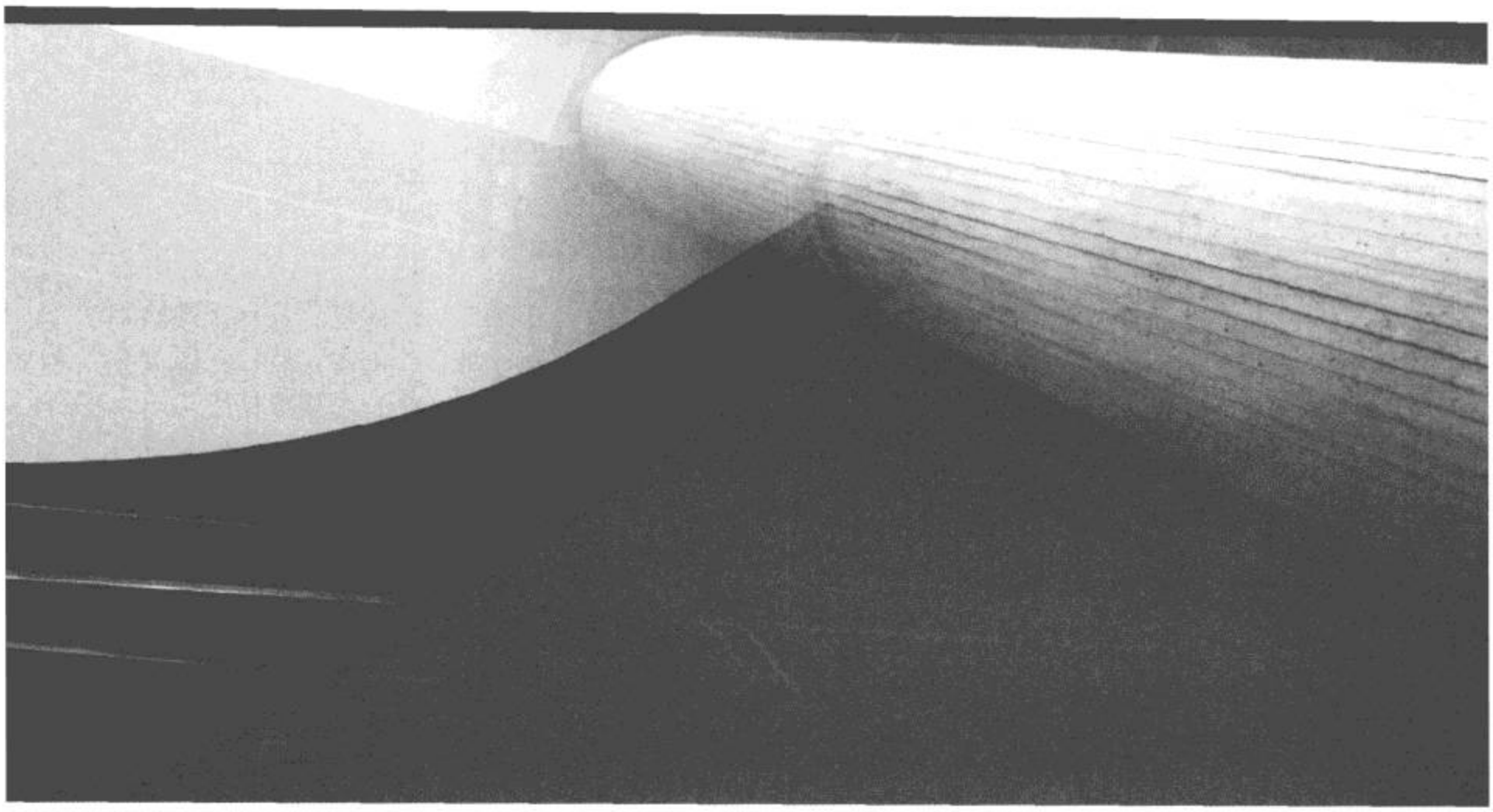
步骤 16 经常执行灾难恢复操作和测试备份操作。我们建议每星期至少对每个数据库执行一次 restore database validate check logical 命令，以确保数据库是可恢复的。

需要注意的是，执行每个步骤时，我们都宁可谨慎过度也不能有丝毫的差错。由您自己来断定 RMAN 的设置是否已完全没有问题。但没有什么比恢复数据库时见到下面的 RMAN 错误消息更令人沮丧的了:

```
RMAN-00571: ===========================================================
RMAN-00569: =========== ERROR MESSAGE STACK FOLLOWS ===========
RMAN-00571: ===========================================================
RMAN-03002: failure of restore command at 07/07/2006 17:14:55
RMAN-06026: some targets not found - aborting restore
```

11.11　小结

本章详细介绍了 RMAN 的备份操作。介绍了如何使用 RMAN 来执行脱机备份操作和联机备份操作，还介绍了 RMAN 的增量备份。讨论了为备份操作配置的默认值以及这些默认值如何使备份进程更加简便，介绍了 backup 命令的一些选项(如使用标记、重写默认的保留策略、强制备份只读的数据文件等)，并且介绍了使用 RMAN 检测数据库损坏的方法。所有这些内容将使您进一步了解 RMAN，并且有助于学习 RMAN 的恢复操作。



第 12 章

RMAN 还原与恢复

我们所有的计划和备份操作都是为了能够还原这些备份和恢复数据库。本章主要介绍还原备份和恢复数据库。本章将首先介绍恢复进程涉及的一些基本问题；然后介绍在磁盘上还原 RMAN 备份以及出现故障时恢复 Oracle 数据库可能需要使用的不同方法。我们强烈建议：如果不大清楚何时需要恢复数据库以及恢复操作所处的环境条件，就应当在测试数据库中测试这些不同类别的恢复操作。

本章将介绍 RMAN 中基本的还原操作。当给定的数据库非常大时，恢复数据库所涉及的情况会相当复杂(我们尝试计算出这些情况，但是没有足够的预算来设计计算机以计算出答案)。您不仅需要理解恢复的机制(本章将作介绍)，而且要理解恢复的机制与 Oracle 的体系结构是如何一起运作的。本章将介绍一些非常简单的恢复环境，第 15 章将涉及一

些更高级的恢复技术，第 21 章将讨论一些不同的恢复案例。建议您阅读本章并测试所介绍的每种恢复，然后再学习第 15 章中的高级恢复技术，一旦熟悉了恢复操作，就可以尝试实现第 21 章中介绍的案例分析。

12.1 RMAN 还原与恢复基础

本章将讨论还原与恢复的概念。在 RMAN 用语中，“还原”与“恢复”具有不同的含义。还原(restore)是指访问先前生成的备份集，从中得到一个或多个对象，然后在磁盘上的某个位置还原这些对象。还原与恢复是分离的。恢复(recovery)是一个使数据库与给定时间点相一致以保证能够打开这个数据库的实际进程，该进程通常是通过应用重做(包括联机重做日志和归档重做日志)来完成的。

Oracle RDBMS 是特定的。数据库中的数据状态对数据库是非常重要的，在数据库启动时要求数据与给定的时间状态一致。如果数据库是一致的，就可以打开这个数据库；如果数据库不一致，就不能打开这个数据库。通过使用回滚段，就可以在启动和运行数据库时保持这种一致性。正常关闭数据库时，数据文件会再次一致。您可以注意到，这里所讨论的主题是一致性。

但是，无数的原因都会导致数据库的不一致性。非正常地关闭数据库(如关闭操作异常中断、服务器掉电等)会使数据库处于不一致的状态中。Oracle 不能接受不一致性，并且在面对不一致性状态时将会拒绝启动数据库。

当然，其他问题也可能导致数据库出现问题。如果磁盘出现故障，就可能丢失数据库数据文件，甚至会丢失整个数据库。由于会发生这些丢失数据库或数据库数据文件的情况，所以我们使用 RMAN 来备份数据库数据文件。问题是当我们要还原这些数据文件时，这些数据文件可能会不一致。更糟的是，这些还原的数据文件可能是几小时前甚至是几天前的数据文件，这样更不能保证其一致性。

正如我们所介绍的，Oracle 严格要求数据的一致性。当我们在恢复数据文件后重新启动数据库时，Oracle 会检测这些数据文件是否状态不一致。Oracle 如何检测数据文件的状态是否一致呢？我们曾经在第 1 章中讨论过 SCN，Oracle 会跟踪每个数据库数据文件头中的当前 SCN 和控制文件中的当前 SCN。打开数据库时，Oracle 要经历 3 个阶段：未加载、加载和打开。数据库启动进入打开阶段时，Oracle 会开始尝试打开这个数据库。在尝试打开数据库的时候，Oracle 检查控制文件中的 SCN 以及每个数据库数据文件中的 SCN，如果这些 SCN 不匹配，Oracle 就会知道数据库存在某些错误并且处于不一致的状态。

启动数据库时还会执行其他检查操作，其中有一些检查操作不涉及一致性的检查。例如，Oracle 会检查每个数据文件中指示该数据文件是否处于热备份模式的一个比特位，如果这个比特位被设置为 1，Oracle 就不会打开数据库。在这种情况下，我们可以简单地执行 alter database datafile end backup 命令来重新启动这个数据库。在尝试打开数据库时，Oracle 还可能判断给定的数据文件是否丢失。如果出现丢失数据文件的情况，最简单的解决方案是加载这个数据文件所在的文件系统，或者还原这个数据文件。

如果发现数据库是不一致的，Oracle 就需要判断是否能够基于联机重做日志来恢复

这个数据库，或者是否需要归档的重做日志来完成恢复操作。如果 Oracle 能够使用联机重做日志来执行恢复操作，它就会执行灾难恢复(或者在 RAC 环境中执行实例恢复操作)。如果 Oracle 不能使用联机重做日志来确保数据库的一致性，它就会停止启动进程，同时报告需要执行介质恢复操作。

如果数据库置于 ARCHIVELOG 模式中，那么 Oracle 所执行的介质恢复操作就需要使用归档重做日志来恢复数据库，并且保证这个数据库的一致性。如果数据库置于 NOARCHIVELOG 模式中，就需要还原/恢复数据库的一致备份。本章除了介绍数据库恢复的有关知识外，还将介绍所有的恢复类型。有些时候我们需要恢复整个数据库，有些时候只需要恢复一些数据文件，本章将讨论这些不同的情况。

上面简要介绍了数据库还原与恢复的一些基础知识，接下来将进一步讨论实际的操作。

12.2　还原数据库前的准备工作

本章将从头介绍还原与恢复进程。这里，我们假定数据库内容全部丢失，只保存了数据库的一些备份。因此，首先介绍服务器参数文件的还原(如果使用了这个文件)，然后再介绍控制文件的还原。有了这些预备文件后，就可以开始还原与恢复数据库。当然，具体的恢复环境也可能不要求还原这些文件，比如您可能只是丢失了某些数据文件，或者只是要执行时间点恢复操作。且某些还原操作需要使用这些预备文件，而另外一些还原操作则不需要使用这些预备文件。对此，我们在本章的章节之间以及后面的案例学习中进行了讨论，相信您会清楚何时需要还原什么文件。

最后，需要注意的是，在大多数示例中，除非使用恢复目录时的操作情况不同，否则我们都不会连接恢复目录，这样可以保证示例的一致性并减少重复性的介绍。因此，如果要使用恢复目录，我们会在“注意”中给出提示。

12.3　RMAN 还原与恢复前的操作

在执行 RMAN 还原与恢复操作前，我们需要执行一些其他操作。要还原数据库，就必须做好准备工作。在尝试执行还原与恢复操作前，您应该与系统管理员一起完成下面的工作：

- 为 Oracle 配置 OS 参数。
- 正确配置磁盘及磁盘大小。
- 安装了磁带驱动器，并且安装和配置了这个磁带所使用的软件。
- 网络能够运作。
- 安装了 Oracle RDBMS 软件。
- 配置了 MML。
- 从备份中恢复一些 RMAN 没有备份的辅助项，这些辅助项包括：
  - 数据库互联文件(例如 sqlnet.ora 文件或 listener.ora 文件)。

- oratab 文件(如果使用了该文件)。
- 数据库参数文件(如果这个文件不是服务器参数文件，并且没有使用 RMAN 备份该文件)。
- 在磁盘上创建的任何 RMAN 控制文件备份(如果启动了控制文件的自动备份操作)。

一旦还原了这些项，就可以开始着手还原 Oracle 数据库。如果要使用恢复目录，就必须首先恢复这个恢复目录，然后才能恢复其他数据库。

启动数据库恢复时，我们必须先恢复服务器参数文件(如果使用了这个文件并且备份了该文件)，然后再恢复控制文件。下面两小节将详细介绍这两个文件的恢复操作。

12.3.1 有关恢复、恢复目录及 MML 层的注意事项

源于在世界范围内讲解 RMAN 的经历，我们学到一些非常重要的事情。其中之一就是当使用某个 MML API(应用程序接口)接口时，有必要使用恢复目录。您可能会问为什么恢复目录如此重要？

记住所有重要的 RMAN 元数据都存储在控制文件中。如果丢失了控制文件，则所有的 RMAN 元数据也将会丢失。在此情况下，如果设置了自动备份控制文件备份(set controlfile autobackup on)，就可以保护控制文件使其免于丢失。但我们经历过一些案例，在这些案例中，从 MML 备份中还原控制文件非常复杂且耗费时间。例如，我们曾见过一个案例，备份成功，但控制文件自动备份却不存在。如果处于这类环境中，一个可用的恢复目录显得非常重要。

在执行还原操作的情况下，若使用了恢复目录将会使相关恢复工作更加容易，将为您节省大量的时间。多数的 MML 供应商不支持从其备份介质中直接取走 RMAN 文件，不支持在不使用 RMAN 客户端的情况下执行还原操作。想要得到包含备份介质及其创建时的时间/日期的 RMAN 备份集片的列表亦非常困难。最后，不能使用 catalog 命令为备份集片创建目录将其备分到 MML 设备上。如果控制文件丢失，需要手动恢复时，对备份集片的恢复将更加复杂(恢复是存在可能的，但会非常复杂)。总体来说，使用恢复目录将会带来许多好处。在第 10 章中已详细讨论了恢复目录。

12.3.2 还原服务器参数文件(SPFILE)

如果没有使用 Oracle 9*i* Release 2 或更新的版本，并且没有使用服务器参数文件，就可以跳过本小节的内容，这时我们只需要从操作系统备份中还原服务器参数文件(或数据库参数文件)。另一方面，如果使用服务器参数文件，并且使用 RMAN 的控制文件自动备份功能备份了服务器参数文件，您就能够从本小节中获取有用的知识。

在第 9 章中，我们曾经提到：如果不使用恢复目录来执行数据库的 RMAN 备份操作，您就可能需要记住用于还原与恢复目的的数据库 DBID。尽管 DBID 不是百分之百地重要，不过它会使还原与恢复操作更加方便。

服务器参数文件的恢复方式有几种，包括不使用闪回恢复区的恢复和使用闪回恢复区的恢复。RMAN 也提供了恢复服务器参数文件的其他方法，下面将详细介绍这些方法。

1. 使用 RMAN(不使用 FRA)从自动备份中恢复服务器参数文件

如果丢失了服务器参数文件，并且使用了自动备份控制文件的功能，我们就可以从控制文件的自动备份中恢复这个服务器参数文件。前面曾经介绍过，根据所使用的操作系统，Oracle 会默认地在$ORACLE_HOME/dbs 或 ORACLE_HOME%\database 目录中备份服务器参数文件与控制文件。如果选择使用默认的位置(或者另一个磁盘位置)备份这些文件，可能需要在另一个备份介质(如磁带)上备份这些备份集(RMAN 可以完成这个功能)。

还原服务器参数文件的一般过程是先设置 ORACLE_SID 和登录 RMAN，然后设置 DBID，使 RMAN 知道需要查找哪一个数据库的服务器参数文件，还必须在这个操作的同时启动数据库实例。通常，Oracle Database 10g 在不存在基本参数文件的情况下不会启动数据库实例。

启动数据库实例后，如果使用默认的位置来存储控制文件的自动备份，就可以简单地执行 restore spfile from autobackup 命令，这样 RMAN 就可以查找包含最新服务器参数文件(SPFILE)备份的控制文件备份集。一旦恢复了服务器参数文件，就必须关闭并重新启动数据库实例，以便应用这个新的服务器参数文件。如果使用的是非默认位置，就需要分配一个指向该位置的信道，然后再使用相同的方法来还原服务器参数文件。

执行 restore spfile from autobackup 命令时，Oracle 会在默认位置中(或者在使用 allocate channel 命令定义的位置中)查找自动的控制文件备份集。由于使用了 set dbid 命令，所以 RMAN 知道数据库的 DBID。RMAN 使用这个 DBID 在定义的目录中查找数据库的最新控制文件备份集。RMAN 创建控制文件自动备份片时，它会使用默认的命名约定。如图 12-1 所示的是一个自动的控制文件备份集片以及相应的命名约定。注意，这种命名约定不适用于 FRA。

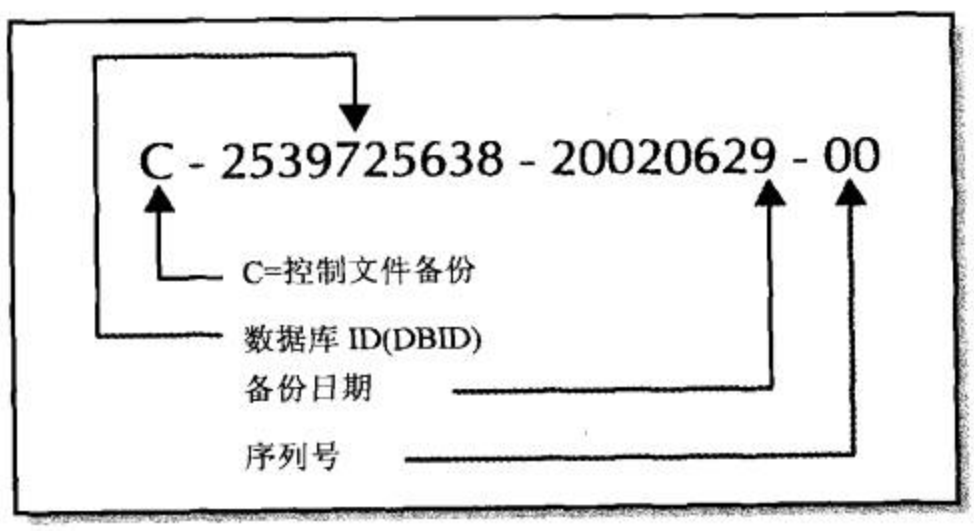


图 12-1　控制文件备份集片使用默认命名约定的图形示例

需要记住的是，如果没有使用恢复目录(并且没有使用控制文件，这是可能的)，Oracle 就不会知道最新控制文件备份片的名称，因此 Oracle 会基于用于这些备份集的默认命名标准来构造控制文件备份片的名称(本章后面部分将作介绍)。Oracle 会遍历整个目录，向后查找数据库的控制文件备份集。在默认情况下，Oracle 会查找一个在过去 10 天内创建的控制文件备份集。如果 Oracle 没有查找到在指定时间周期内创建的控制文件备份集，

就会产生一个错误。如果 Oracle 发现一个有效的控制文件备份集，就会为用户还原服务器参数文件。

使用 restore 命令的 maxseq 和 maxdays 参数可以修改 RMAN 向后查找控制文件自动备份的时间周期。下面是一组恢复控制文件的命令示例，首先我们使用的是 restore 命令的默认设置：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
restore spfile from autobackup;
shutdown immediate;
```

当然，我们可能不希望使用默认值，此时，假定我们在 restore 命令中定义自动备份的默认位置。下面是这种类型操作的示例：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 206232339;
startup nomount;
run
{
set controlfile autobackup format for device type disk
to 'c:\oracle\backup\%F';
allocate channel c1 device type disk;
restore spfile from autobackup;
}
shutdown immediate;
```

接下来使用 maxseq 和 maxdays 参数在超过默认天数(10 天)的时间周期内向后查找控制文件备份集：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
run
{
set controlfile autobackup format for device type disk
to 'c:\oracle\backup\%F';
allocate channel c1 device type disk;
restore spfile from autobackup maxseq 200 maxdays 100;
}
shutdown immediate;
```

将服务器参数文件还原到不同的位置和文件中并不是非常复杂。在下面的示例中，假定数据库正在启动和运行，因此不需要指示自动备份文件集的位置。

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
restore spfile to 'c:\oracle\spfile.restore' from autobackup;
shutdown immediate;
```

2. 使用 RMAN 和 FRA 从自动备份中恢复服务器参数文件

使用 FRA 恢复服务器参数文件的过程实际上比不使用 FRA 的恢复过程简单。首先，我们不需要担心 DBID 没有意义也不用操心分配信道的工作，这就节省了一定的时间。恢复服务器参数文件时，我们只需要正确配置临时参数文件(确保适当地设置 FRA 目标目录)，启动 Oracle 实例，并且执行 restore spfile 命令。下面是一个示例：

```
set oracle_sid=recover
rman target sys/robert
startup nomount;
restore spfile from autobackup;
shutdown immediate;
```

我们仍然可以使用 maxseq 和 maxdays 参数恢复数据库参数文件，如下面的代码段所示：

```
restore spfile from autobackup maxseq 200 maxdays 100;
```

3. 从特定的备份集中恢复服务器参数文件

如果使用过 Oracle Database 10g 之前的 RMAN 版本，就知道如果没有配置自动备份，恢复服务器参数文件会非常困难。在这种情况下，需要编写一些 PL/SQL 代码来从 RMAN 备份中还原服务器参数文件。

Oracle Database 10g 简化了这个过程。现在，restore 命令包含了一个 from backup 子句，使用该子句可以指示包含服务器参数文件备份的备份集片。因此，只需要知道该备份集片的名称和它的位置，就可以还原服务器参数文件。在某些情况下，查找包含服务器参数文件备份的备份集片可能存在一些困难，但一旦找到了该备份集片，还原操作就会非常简单(当然，最好是还原最新的控制文件备份)。

为了还原服务器参数文件，首先启动数据库实例并标识包含服务器参数文件备份的备份集片(或者进行猜测)。接着启动 RMAN。然后设置数据库的 DBID(前面已经讨论过如何设置)。一旦设置了 DBID，就需要分配信道给要还原的设备。分配信道后，执行 restore spfile from 命令，在命令的末尾传入备份集片的名称。于是 RMAN 将服务器参数文件还原到正确的位置。下面是执行这种类型还原的示例：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
run
```

```
{
  allocate channel c1 device type disk;
  restore spfile
  from 'd:\backup\recover\C-2539725638-20060629-00';
}
shutdown immediate;
```

现在，可以执行一些其他的还原活动。

4. 使用恢复目录恢复服务器参数文件

如果使用了恢复目录，在启动(未加载)Oracle 实例后简单地执行 restore spfile 命令就可以还原最新的服务器参数文件。RMAN 会使用恢复目录找出最新的控制文件备份，并析取这个备份以供用户使用。下面是一个示例：

```
set oracle_sid=recover
rman target sys/robert catalog rcat_manager/password@robt
startup nomount;
restore spfile from autobackup;
shutdown immediate;
```

需要注意的是，还原后要关闭数据库，这样才能确保重启时使用的是正确的参数值设置。

5. 使用联机操作的数据库还原备份的服务器参数文件

不管是否使用控制文件或恢复目录，在数据库启动时从数据库备份中析取服务器参数文件的副本是非常容易的。我们应当注意到这个操作会生成一个文本参数文件，而不是生成一个服务器参数文件，因此如果要将这个文本参数文件用作服务器参数文件，就必须转换该文件。

如果不是使用恢复目录并且控制文件的自动备份已启用，就只需要执行下面的命令：

```
RMAN> restore spfile to pfile 'd:\backup\test.ora' from autobackup;
```

执行这条命令会在 d:\backup 目录中将服务器参数文件还原为 test.ora 文件。对于任何自动备份的还原操作来说，如果不提供 maxseq 和 maxdays 参数值，RMAN 就只需要查找过去 7 天内生成的控制文件的自动备份片。

如果没有使用恢复目录与控制文件自动备份，或者是在使用恢复目录，就可以执行下面的命令：

```
RMAN> restore spfile to pfile 'd:\backup\test.ora';
```

在这种情况下，Oracle 会使用数据库的控制文件找到最新的备份集，并从这个备份集中还原服务器参数文件。这样，我们就还原了服务器参数文件，接下来将还原控制文件了。

12.3.3　还原控制文件

还原控制文件与还原服务器参数文件的差异不是很大。我们首先需要加载数据库，因此要还原控制文件，就应当已经还原或创建了服务器参数文件并且加载了数据库。一旦满足了这些要求，就可以开始恢复数据库的控制文件。下面我们将详细介绍恢复控制文件的进程。

恢复控制文件有多种方式，包括不使用 FRA 的恢复和使用 FRA 的恢复。在下面的部分中将介绍额外的控制文件恢复情况。

1. 使用 RMAN(不使用 FRA)从自动备份中恢复控制文件

如果丢失了控制文件，并且没有使用 FRA，就会希望从控制文件的自动备份集中恢复这个控制文件(强烈推荐使用这种功能)。前面曾经介绍过，根据所使用的操作系统，Oracle 会默认地在$ORACLE_HOME/dbs 或$ORACLE_HOME/database 目录中备份控制文件(以及服务器参数文件)。如果选择使用默认的位置备份这个文件(或者备份到可选的磁盘位置)，就可能需要在另一个备份介质(如磁带)上备份这些备份集(RMAN 可以完成这个功能)。

还原控制文件时，假设已经还原了实例参数文件并且启动了该实例(例如，使用 startup nomount 命令)。一旦实例启动运行，在没有使用 FRA 的情况下，还原控制文件的一般过程就是先设置 ORACLE_SID 和登录 RMAN(如果还没有登录)。和服务器参数文件的还原一样，还需要设置 DBID，使 RMAN 知道需要查找哪一个数据库的控制文件。

现在已经准备好还原控制文件。和服务器参数文件一样，如果没有使用默认的位置(很少是这种情况)来存储控制文件的自动备份，就需要执行 restore control file from autobackup 命令，RMAN 就会查找包含最新控制文件备份的控制文件备份集。一旦恢复了控制文件，就需要关闭该实例并重新启动，从而使新的控制文件参数生效。如果使用的是非默认位置，则使用 allocate channel 命令来分配指向该位置的信道，然后就可以使用相同的方法还原控制文件。

执行 restore controlfile from autobackup 命令时，Oracle 会在默认位置中(或者在使用 allocate channel 命令定义的位置中)查找自动的控制文件备份集。由于使用了 set dbid 命令，所以 RMAN 知道数据库的 DBID。RMAN 使用这个 DBID 在包含控制文件备份的目录中查找数据库的最新控制文件备份集(查看前面关于服务器参数文件恢复的部分，可了解 RMAN 用于控制文件自动备份备份集的命名约定的格式)。和服务器参数文件的恢复相同，通过应用 restore 命令的 maxseq 和 maxdays 参数，可以修改 RMAN 查找控制文件自动备份的时间周期。

下面是一组恢复控制文件的命令示例，首先使用 restore 命令的默认设置：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
restore controlfile from autobackup;
```

```
shutdown immediate;
```

接下来再使用 maxseq 和 maxdays 参数在超过默认天数(10 天)的时间周期内向后查找控制文件备份集。这一次假定没有备份到默认的位置：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
run
{
set controlfile autobackup format for device type disk
to 'c:\oracle\backup\%F';
allocate channel c1 device type disk;
restore controlfile from autobackup maxseq 200 maxdays 100;
}
shutdown immediate;
```

将控制文件还原到不同的位置和文件中并不非常复杂。在下面的示例中，假定数据库已启动并运行，因此不需要定义控制文件自动备份的位置：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
run
{
 restore controlfile to 'c:\oracle\control file.restore' from
 autobackup;
}
shutdown immediate;
```

2. 使用 RMAN 和 FRA 从自动备份中恢复控制文件

和服务器参数文件的恢复一样，使用 FRA 恢复控制文件的进程非常简单。首先，确保定义 FRA 的位置的参数被正确设置。然后，启动数据库实例；执行 restore controlfile from autobackup 命令即可恢复控制文件。下面是一个示例：

```
set oracle_sid=recover
rman target sys/robert
startup nomount;
restore controlfile from autobackup;
shutdown immediate;
```

仍然可以使用 maxseq 和 maxdays 参数进行恢复，如下面的代码段所示：

```
restore controlfile from autobackup maxseq 200 maxdays 100;
```

3. 恢复较早的控制文件备份

有时，可能要恢复的是一段时间以前建立的控制文件备份，因此控制文件不是来自于最新的备份(执行 resetlogs 命令时通常是这种情况)。结合 until time 参数，restore control file from autobackup 命令可以实现这种恢复操作。例如，通过执行如下命令，可以查找并还原在 2005 年 12 月 9 日的早上 4:11 时或之前创建的控制文件备份：

```
restore controlfile from autobackup until time
"to_date('12/09/2005 13:00:00','MM/DD/YYYY HH24:MI:SS')";
```

如果要使用这个命令，就需要知道哪些控制文件备份可用于还原操作。在第 18 章中将更详细地讨论 list 命令，现在只需知道使用 list backup of controlfile 命令可以确定哪些控制文件备份可用于还原操作。下面是一个示例：

```
RMAN> list backup of controlfile;
List of Backup Sets
===================
BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
38      Full    6.80M      DISK        00:00:05     01-JAN-06
  BP Key: 39 Status: EXPIRED Compressed: NO Tag: TAG20060108T183828
       Piece Name: C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\AUTOBACKUP\2006_01_08\O1_MF_S_579206308_1W3D485F_.BKP
 Control File Included: Ckp SCN: 3568285 Ckp time: 08-JAN-06

BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
40      Full    6.80M      DISK        00:00:06     08-JAN-06
   BP Key: 41 Status: AVAILABLE Compressed: NO Tag: TAG20060108T183857
       Piece Name: C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\AUTOBACKUP\2006_01_08\O1_MF_S_579206337_1W3D5495_.BKP
  Control File Included: Ckp SCN: 3568311 Ckp time: 08-JAN-06
```

在前面的示例中，使用 list 命令可以发现有两个不同的控制文件备份。根据当前的备份保留策略，第一个控制文件备份(BS Key 38)的状态为 EXPIRED。因为该备份已经过期，RMAN 不会在执行还原操作期间使用它(如果备份集片仍然可用，则可以使用 change 命令改变它的状态，从而能够使用它)。第二个控制文件备份(BS Key 40)可用，RMAN 将用它还原这个备份。

注意，list 命令只有在使用最新的控制文件加载数据库时可用。如果没有可用的控制文件，则 list 命令就会无效。此外，如果没有可用的控制文件，则可以忽视保留标准，使用 autobackup 命令可还原任何控制文件。

3. 从备份集中恢复控制文件

如果使用过 Oracle Database 10g 之前的 RMAN 版本，就会知道如果没有配置自动备份，恢复控制文件就可能非常困难。在这种情况下，需要编写一些 PL/SQL 代码来还原

控制文件。

Oracle Database 10g 的控制文件恢复操作更为简单。现在，restore 命令包含了一个 from backup 子句，使用该子句可以指定包含控制文件备份的备份集片。因此，只需要知道该备份集片的名称和它的位置，就可以还原控制文件。在某些情况下，查找包含控制文件备份的备份集片可能存在一些困难，但一旦找到了该备份集片，还原操作就会非常简单(当然，最好是还原最新的控制文件备份)。

为了还原控制文件，首先启动数据库实例并标识包含控制文件的备份集片(或者进行猜测)。接着启动 RMAN。然后设置数据库的 DBID(前面已经讨论过如何设置)。一旦设置了 DBID，就需要分配信道给要还原的设备。分配信道后，执行 restore controlfile from 命令，在命令的末尾传入备份集片的名称。于是 RMAN 将还原控制文件并将其复制到不同的 CONTROL_FILE 参数位置。下面是执行这种还原类型的示例：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
run
{
  allocate channel c1 device type disk;
  restore controlfile
  from 'd:\backup\recover\C-2539725638-20020629-00';
}
shutdown immediate;
```

现在，就可以执行一些其他的还原活动。

4. 使用恢复目录恢复控制文件

如果使用了恢复目录，可以简单地执行 restore controlfile 命令，还原最近的控制文件备份。RMAN 会使用恢复目录找出最近的控制文件备份，并析取这个备份以供用户使用。下面是一个示例：

```
set oracle_sid=recover
rman target sys/robert catalog rcat_manager/password@robt
# Note - We would issue a startup nomount
# and restore control file from autobackup here if we needed to.
# shutdown immediate here if we recovered the control file.
startup nomount;
restore controlfile;
# mount the database in preparation for a restore.
alter database mount;
```

5. 还原控制文件后的数据库恢复

当使用 RMAN 还原控制文件时(并且没有丢失数据文件或联机重做日志)，有两件事情需要考虑。首先，需要考虑在还原控制文件后如何启动和运行数据库。其次，需要恢

复与 RMAN 相关联的信息，例如使用 RMAN 在新创建的控制文件中注册的归档重做日志和备份集。接下来详细介绍这两方面的内容。

在恢复控制文件后打开数据库

如果只丢失了控制文件，那么恢复控制文件和数据库将就非常直观(如果某类型的 Oracle 恢复操作可以被认为直观)。恢复控制文件的命令是相同的，只需模拟一个不完整恢复操作以打开数据库。注意， 在一些情况下可能需要在完成恢复之前注册归档的重做日志或备份集。在下一节中将讨论这个注册进程。

如果在 NOARCHIVELOG 模式下运行，则需要使用下面的恢复方法(在该示例中，假定不使用 FRA，因此设置 DBID)：

```
set oracle_sid=recover
rman target sys/robert
set DBID = 2539725638;
startup nomount;
restore controlfile from autobackup;
alter database mount;
recover database noredo;
alter database open resetlogs;
```

如果是在 ARCHIVELOG 模式下运行，如下所示，恢复方法会稍有不同(在该示例中，假定使用 FRA，因此不设置 DBID)：

```
set oracle_sid=recover
rman target sys/robert
startup nomount;
restore controlfile from autobackup;
alter database mount;
recover database;
alter database open resetlogs;
```

当然，丢失其他数据库物理文件可能会使这种恢复操作变得复杂。此外，如果没有最近的控制文件备份，则操作就可能变得非常复杂，并且可能最终必须使用 create control file 命令手动重新创建控制文件。我们希望您永远都不会面对这种糟糕的情况，因为如果备份和恢复策略非常合理就不会遇到这种情况。

还原到 RMAN 相关的记录的控制文件中

一旦还原了控制文件，就需要考虑丢失 RMAN 相关数据的可能性。由于 RMAN 将所有数据存储在控制文件中，很有可能至少丢失与归档的重做日志相关的一些记录(Oracle 需要完全恢复归档的重做日志)。

RMAN 简化了这个问题的处理，因为可以在还原控制文件后注册各种 RMAN 相关的备份文件，从而确保控制文件中有恢复所需的全部文件。为了重新创建 RMAN 相关的归档重做日志的记录，可以使用 RMAN 的 catalog 命令来在控制文件中注册归档的重做日志。catalog 命令可用于注册特定的备份集片，如下所示。

```
RMAN>Catalog backuppiece 'c:\oracle\product\10.2.0\flash_recovery_area
\testoem\backupset\2005_12_09\O1_MF_ANNNN_TAG20051209T041150_
1SLP386H_.BKP';
```

也可以对归档的重做日志编写目录，如下所示：

```
RMAN>Catalog archivelog
'c:\oracle\product\10.2.0\flash_recovery_area\testoem\archivelog\
2005_12_15\O1_MF_1_2_1T3SVF05_.ARC';
```

但如果一天中生成 1000 个归档的重做日志时该怎么办呢？手动为它们编写目录吗？幸运的是，RMAN 开发人员想到了这一点。使用 RMAN 可以对整个目录编写目录，而不需要列出单个的文件。只要再次使用 catalog 命令以及以下的一个关键字即可：

- recovery area 或 db_recovery_file_dest
- start with

recovery area 和 db_recovery_file_dest 关键字的功能相同，它们使 RMAN 对整个 FRA 编写目录。如果 RMAN 发现已经对一些文件编写目录，它就会跳过这些文件并且继续对没有在控制文件中找到的剩余文件编写目录。下面是对 FRA 中的所有文件编写目录的一个示例：

```
RMAN> catalog recovery area;
```

如果没有使用 FRA，则可以使用 start with 语法。start with 语法允许遍历非 FRA 的备份目录并且对包含在该目录以及该目录下任何子目录中的所有 RMAN 相关文件编写目录。下面是使用 catalog start with 命令的一个示例：

```
catalog start with 'c:\oracle\backups\testoem';
```

注意：
如果使用备份控制文件执行还原操作，Oracle Database 10g R2 中的 RMAN 就会自动对 FRA 编写目录。

6. 联机还原控制文件

不管是使用控制文件还是恢复目录，在数据库启动时从数据库备份中析取控制文件的副本是非常容易的。

如果不使用恢复目录而要启用控制文件的自动备份，就只需要执行下面的命令：

```
RMAN> restore controlfile to 'd:\backup' from autobackup;
```

执行这条命令会在 d:\backup 目录中将服务器参数文件还原为 test.ora 文件。对于任何自动备份的还原操作来说，RMAN 只默认查找过去 7 天内生成的控制文件自动备份片。可以使用 maxseq 和 maxdays 参数修改这个默认的时间周期。

如果没有使用恢复目录与控制文件自动备份，或者使用了恢复目录，则可以执行下面的命令：

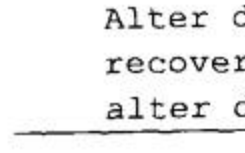
```
RMAN> restore controlfile to 'd:\backup';
```

在这种情况下，Oracle 会使用数据库的控制文件找到最近的备份集，并从这个备份集中还原控制文件。当然，可以使用 dbms_backup_restore 进程(本章前面介绍过)来手动还原控制文件。

RMAN 实习：恢复控制文件

实习注意事项

在这个实习中，需要一个 Oracle 安装软件和一个可操作的 Oracle 测试数据库，还要配置 FRA 并对该区域进行备份。



注意：
本实习中，数据库置于 ARCHIVELOG 模式下。

步骤 1　确保配置了控制文件的自动备份操作：

```
configure controlfile autobackup on;
```

需要注意的是，在这种情况中，将在默认的位置中创建控制文件备份集片。

步骤 2　完成系统的备份(这时假设是热备份)。在这个实习中，我们假定在配置的默认设备上创建备份：

```
set oracle sid=recover
rman target rman_backup/password
backup database plus archivelog;
```

步骤 3　使用 shutdown immediate 命令关闭数据库。在这个实习中不要使用 shutdown abort 命令：

```
shutdown immediate;
```

步骤 4　重命名数据库控制文件的所有副本。不要删除这些副本，以免不能恢复备份。

步骤 5　启动数据库，不过此时无法查找到控制文件，并且无法打开它。

```
startup;
```

步骤 6　使用控制文件的自动备份通过 RMAN 恢复控制文件：

```
restore controlfile from autobackup;
```

步骤 7　加载数据库，必须通过模拟不完全恢复操作来完成整个恢复进程：

```
Alter database mount;
recover database;
alter database open resetlogs;
```

12.4　restore 和 recover 命令

恢复数据库的基本过程分为两个步骤。第一个步骤是使用 restore 命令还原数据库备份，第二个步骤是使用 recover 命令恢复数据库，包括应用归档的重做日志。在使用这些命令恢复数据库之前，先详细介绍这两个命令。

12.4.1　restore 命令

虽然 restore 命令还有一些辅助目标，但其主要功能是从 RMAN 备份中还原文件，为恢复做准备。RMAN 和 restore 命令非常智能，它们将选择最近的备份来进行还原，从而减少恢复时间。其结果是，restore 命令可能从备份集中还原数据文件，也可能从映像副本中还原数据文件，或者两者并用来进行还原，只要对加快还原过程有帮助即可。

restore 命令用于从自动的备份中还原服务器参数文件和控制文件。restore 命令也可用于创建备用数据库的备用控制文件。我们也使用 restore 命令来将数据库还原到任何时间点。在这种情况下，该命令将查找离给定时间点最近的备份以进行还原。如果没有恢复目录，RMAN 可以在数据库的对应物中将数据库还原到任何时间点(假设备份可用)。restore 命令也可从前面的对应物中还原数据库，但需要在建立对应物期间备份控制文件。如果使用恢复目录，就可以将数据库还原到任何对应物。

restore 命令也可根据分配给备份的给定标记还原特定的备份。如果开发环境中存在需要经常用于还原的“黄金”备份，这种还原方式就非常有用。如果需要归档的重做日志(如 LogMiner 的操作)，restore 命令也可用于还原归档的重做日志(但是 recover 命令将在数据库恢复期间完成该操作)。

此外，restore 命令可用于验证实际恢复数据库的能力。这将确保备份可用于还原数据库，并且可以验证这些备份的完整性。也可以使用 restore preview 命令标识还原数据库所需的备份(非常类似于将在第 18 章中讨论的 list 命令)。

注意：
如果您想要从 RMAN 无法知悉的存储设备上获取备份集时，使用 restore preview 命令可以提供方便。例如，如果您使用 RMAN 将备份集备份到磁盘，然后使用操作系统备份这些文件到磁带，RMAN 就无法知悉备份到磁带的文件情况。这时可以应用 restore preview 命令确定需要还原哪些使用操作系统备份的文件，进而简化了还原过程。

使用 restore 命令时，如果为备份加密，则需要确保加密方法的正确配置。因此，如果使用透明模式的加密，相应的电子皮夹必须可用(后面将讨论加密)。

使用 restore 命令时，该命令会在没有任何提醒的情况下重写已经存在的任何文件，除非使用 set newname 命令(本章后面将讨论该命令)。由于这一点，在还原文件时必须非常小心，并且确保重写已经存在的文件不会产生任何不利的影响。

restore 命令也具有故障恢复功能。在恢复期间，如果 RMAN 发现给定的备份文件不可用或者已经被破坏，它就会自动尝试使用前面的备份来完成恢复过程。进行故障恢复时，RMAN 会在数据库警报日志中添加一条相应的消息。

12.4.2　recover 命令

recover 命令用于恢复数据库。该命令可以执行数据库的完全恢复或者时间点恢复。recover 命令确定需要哪些归档重做日志，并且析取和应用它们。一旦完成重做的应用，我们就只需要使用 alter database open 命令打开数据库即可。

recover 命令也用于判定是否存在可应用的增量备份映像。这些映像可应用于基本增量备份或数据文件映像副本。recover 命令总是尝试首先使用增量备份(如果有可用的增量备份)，因为这将是还原数据库最快的方法(与应用归档的重做日志相比)。

还原归档的重做日志时，recover 命令尝试使用磁盘上已有的重做日志。如果磁盘上没有可用的重做日志，recover 就会尝试从各种归档的重做日志备份集中还原它们。注意，可以使用 recover 命令中的 noredo 参数来指示 RMAN 不应该尝试将重做应用于数据库。您将在本章后面的示例中看到，noredo 参数用于 NOARCHIVELOG 数据库的恢复。

12.5　在 NOARCHIVELOG 模式中还原和恢复数据库

如果数据库置于 NOARCHIVELOG 模式下，我们将从完全的脱机备份中恢复这个数据库，并且不可能实现时间点恢复。如果数据库置于 ARCHIVELOG 模式中，请您阅读本章 12.6 节中的内容。如果执行 NOARCHIVELOG 模式数据库的增量备份操作，请您阅读本章 12.6.4 节中的内容。

12.5.1　还原的准备工作

如果在 NOARCHIVELOG 模式中运行数据库并且假定拥有数据库的一个备份，就可以非常容易地执行数据库的完全恢复。首先，要清理所有旧的数据文件，以及联机重做日志和控制文件。虽然并不是非得这样做，但由于使用了 NOARCHIVELOG 模式，我们会希望一切从头开始(当然，为了以防万一，最好还是在某个位置上做好这些文件的备份)。

清理完数据文件、控制文件和重做日志后，就可以开始启动恢复进程。如本章前面所述，首先，可以从最近生成的备份中恢复控制文件，或者可以使用在某个时间点(备份完成后的时期内)创建的备份控制文件。如果使用 create control file 命令，就需要在还原数据库之前把与 RMAN 备份相关的文件编成目录。

在这个示例中，我们假设没有使用恢复目录，并且假设按 RMAN 默认设置从最新的备份中恢复控制文件。使用 set time 命令可以从旧的备份中恢复控制文件，我们将在本节的后面部分讨论这个问题。

如果恢复了服务器参数文件和控制文件，就可以忽视使用恢复目录和不使用恢复目录之间的差异。因此，我们只介绍不使用恢复目录的恢复操作。此外，使用 FRA 和不使

用 FRA 的恢复操作也没有很大的区别。在下面的示例中，我们将使用 FRA 并会强调使用 FRA 时所引发的问题。

如下所示，首先介绍执行恢复操作所使用的 RMAN 命令：

```
startup mount;
restore database;
recover database noredo;
alter database open resetlogs;
```

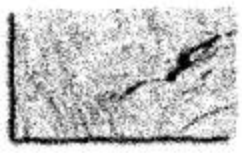
这些 RMAN 命令看上去非常简单。当然，这些步骤假定已经恢复了服务器参数文件和数据库控制文件。第一条命令 startup mount 加载数据库，因此 Oracle 会读取准备用于数据库还原操作的控制文件。第二条命令 restore database 将使 RMAN 启动数据库数据文件的还原操作。接下来是 recover database noredo 命令，该命令指示 RMAN 执行最后的恢复操作以准备打开这个数据库。由于这个数据库在 NOARCHIVELOG 模式中，不应用任何归档的重做日志并且丢失了联机重做日志，所以要求在这条命令中使用 noredo 参数。如果存在完整的联机重做日志，就不需要使用 noredo 参数。最后，使用 alter database open resetlogs 命令打开这个数据库。由于已经还原了控制文件，并且需要重新构建联机重做日志，所以必须使用 resetlogs 命令。实际上，我们可能会在每个 NOARCHIVELOG 模式的恢复操作中都使用 resetlogs 命令。

下面介绍了一个实际运作的恢复操作：

```
d:>set oracle_sid=testoem
d:>rman target sys/robert
RMAN> startup mount
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area 209715200 bytes
Fixed Size 1248164 bytes
Variable Size 100664412 bytes
Database Buffers 104857600 bytes
Redo Buffers 2945024 bytes
RMAN> restore database;
Starting restore at 26-DEC-05
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to
C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\SYSTEM01.DBF
restoring datafile 00002 to
C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\UNDOTBS01.DBF
restoring datafile 00003 to
C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\SYSAUX01.DBF
restoring datafile 00004 to C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\USERS01.DBF
restoring datafile 00005 to
C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\CATALOG01.DBF
```

```
channel ORA_DISK_1: reading from backup piece
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM\BACKUPSET
\2005_12_26\O1_MF_NNNDF_TAG20051226T085336_1V00ZL3Y_.BKP
channel ORA_DISK_1: restored backup piece 1
piece handle=C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM\
BACKUPSET\2005_12_26\O1_MF_NNNDF_TAG20051226T085336_1V00ZL3Y_.BKP
tag=TAG20051226T085336
channel ORA_DISK_1: restore complete, elapsed time: 00:03:26
Finished restore at 26-DEC-05
RMAN> recover database;
Starting recover at 26-DEC-05
using channel ORA_DISK_1
Finished recover at 26-DEC-05
RMAN> alter database open;
```

这样，我们就能够恢复数据库了！



注意：
联机重做日志无效时，可以使用 recover database noredo 命令。联机重做日志在恢复期间有效时，可以使用没有 noredo 参数的 recover database 命令。

12.5.2　在不同的位置上还原数据库

当然，我们并不总是能够将数据库还原回驻留 Oracle 文件的原始文件系统中。例如，在一个灾难恢复练习中，我们可能将数据库恢复到一个大的文件系统中，而不是恢复到 6 个较小的文件系统中。由于 RMAN 默认地将数据文件还原到备份时数据文件所在的原始位置，所以这样将出现问题。怎样解决这个问题呢？

输入 set newname for datafile 和 switch 命令即可。这些命令与 restore 和 recover 命令结合使用时将告知 RMAN 数据文件存放的位置。针对数据库数据文件的重定位，set newname 命令提供了几个选项。在 Oracle Database 10g 及以前的版本中，可以为独立的数据文件设置新的文件名。而在 Oracle Database 11g 中，提供的新特性具有修改表空间中的所有数据文件或者整个数据库中的所有数据文件的位置的能力。

例如，如果数据文件最初被备份到 d:\oracle\data\recover 目录中，而现在要将这些数据文件还原到 e:\oracle\data\recover 目录中，就可以先对每个数据文件执行 set newname for datafile 命令，同时说明这个数据文件的原始位置和新位置。下面是一个使用这条命令的示例：

```
run{set newname for datafile 'd:\oracle\data\recover\system01.dbf'
    to 'e:\oracle\data\recover\system01.dbf';
```

这个示例对于使用 RMAN 时使用的任何 Oracle 版本都适用。需要注意的是，我们同时定义了数据文件的原始位置和 RMAN 应当将文件复制到的新位置。一旦对所有要还原到不同位置的数据文件都执行了 set newname for datafile 命令，就可以继续使用前面所介绍的 restore database 和 recover database 命令。最后，在打开数据库之前，我们必须告

知 Oracle 以后永久地使用已还原的、重定位的数据文件，对此我们要用 switch 命令。

使用 switch 命令可以修改数据库控制文件中数据文件的位置，以反映 Oracle 数据库数据文件的新位置。通常，使用 switch datafile all 命令指示 Oracle 转换控制文件中的所有数据文件位置。此外，使用 switch datafile 命令可以转换特定数据文件的位置。

如果使用 set newname for datafile 命令但不转换所有还原的数据文件，RMAN 就会将没有转换的数据文件当作数据文件的副本，并且在恢复数据库时 RMAN 不会尝试使用这些没有转换的数据文件。下面是一个使用 set newname for datafile 命令执行还原操作的命令示例：

```
startup nomount
restore controlfile from autobackup;
alter database mount;
run
{
set newname for datafile 'd:\oracle\oradata\recover\system01.dbf' to
'e:\oracle\oradata\recover\system01.dbf';
set newname for datafile
'd:\oracle\oradata\recover\recover_undotbs_01.dbf' to
'e:\oracle\oradata\recover\recover_undotbs_01.dbf';
set newname for datafile 'd:\oracle\oradata\recover\users01.dbf' to
'e:\oracle\oradata\recover\users01.dbf';
set newname for datafile 'd:\oracle\oradata\recover\tools01.dbf' to
'e:\oracle\oradata\recover\tools01.dbf';
set newname for datafile 'd:\oracle\oradata\recover\indx01.dbf' to
'e:\oracle\oradata\recover\indx01.dbf';
restore database;
recover database noredo;
switch datafile all;
alter database open resetlogs;
}
```

需要注意的是，如果恢复操作不成功但是却成功地还原了数据文件，这些还原的数据文件就会成为数据文件副本并且不会被删除。

若使用 Oracle Database 11g，通过使用带有 for database 命令的 set newname 命令，在此命令中重命名所有的数据库文件，将会使还原工作更加容易。如果只希望重命名与给定表空间关联的数据文件时，也可以使用 set newname for tablespace 命令。

要将这些新的 set newname 命令关联起来，为避免在移动数据文件时发生文件名冲突的情况，需要使用替换变量。相关的替换变量见表 12-1。

表 12-1　set newname 命令的替换变量

变　量	含　义
%b	使用该替换变量将产生没有任何目录路径信息的文件名
%f	使用该替换变量将为数据文件建立绝对文件号
%U	使用该替换变量将由系统建立保证具有唯一性的文件名

(续表)

变　量	含　义
%I	使用该替换变量将产生数据库的 DBID
%N	使用该替换变量将产生表空间名

下面的示例使用 set newname for database 命令实现对数据库中所有数据文件的重命名。

```
RUN
{
shutdown abort;
startup mount;
SET NEWNAME FOR DATABASE TO 'C:\oradata1\%b';
Restore database;
Recover database;
switch datafile all;
Alter database open;
}
```

若仅需要对特定表空间的数据文件重命名，可以对 set newname 命令做些调整，如下例所示：

```
RUN
{
shutdown immediate;
startup mount;
SET NEWNAME FOR TABLESPACE user_data TO
'c:\oradatanew\users\user_data%b.dbf';
Restore database;
switch datafile all;
Recover database;
Alter database open;
}
```

RMAN 实习：恢复 NOARCHIVELOG 模式的数据库

实习注意事项

在这个实习中，需要一个 Oracle 安装软件和一个可操作的 Oracle 测试数据库。

注意：

对于这个实习来说，数据库在 NOARCHIVELOG 模式中运行。

步骤 1　设置 ORACLE_SID，然后登录 RMAN。确保配置了控制文件的自动备份操作。由于这是一个脱机备份，所以必须关闭和加载这个数据库：

```
set oracle_sid=recover
rman target rman_backup/password
configure controlfile autobackup on;
shutdown immediate;
startup mount;
```

需要注意的是，在这种情况下，我们会在默认的位置中创建控制文件备份集片。

步骤 2 完成系统的冷备份。在这个实习中，我们假定将在配置的默认设备上创建数据库的备份:

```
backup database;
```

步骤 3 关闭数据库:

```
shutdown immediate;
```

步骤 4 重命名所有数据库数据文件、联机重做日志和控制文件(如果没有足够的空间或出现了问题时，您可以承担得起丢失数据库的损失，也可以选择删除这些文件)。

步骤 5 启动和加载数据库，并且还原控制文件:

```
startup nomount;
set DBID = <enter the DBID of the database here>;
restore controlfile from autobackup;
alter database mount;
```

步骤 6 RMAN 使用步骤 2 中生成的备份来恢复这个数据库:

```
restore database;
recover database noredo;
alter database open resetlogs;
```

步骤 7 通过再次备份数据库完成数据库的恢复操作:

```
shutdown immediate;
startup mount;
backup database;
```

注意:
如果联机重做日志没有被删除，可以使用 recover database 命令来代替 recover database noredo 命令。

12.6 在 ARCHIVELOG 模式中恢复数据库

通常，由于一个或多个需求，生产数据库通常会在 ARCHIVELOG 模式中。这些需求包括:

- 时间点恢复
- 为用户提供最小恢复时间服务等级协议(service-level agreement，SLA)
- 执行联机数据库备份操作的能力
- 在数据库有效时恢复指定数据文件的能力

数据库置于 ARCHIVELOG 模式时，我们可以选择下面这些恢复选项：

- 完全数据库恢复
- 表空间恢复
- 数据文件恢复
- 不完全数据库恢复
- 联机块介质恢复

我们将在本节介绍前 3 种恢复选项，本章的后面将介绍不完全数据库恢复。在第 15 章中将详细讨论联机块介质恢复。这些恢复类型会使您发现 ARCHIVELOG 模式中的恢复操作与 NOARCHIVELOG 模式中的恢复操作的最大区别在于归档重做日志的应用，以及执行不完全恢复操作时需要定义的一些问题。现在，先介绍 ARCHIVELOG 模式下的完全数据库恢复。

注意：
不管是否是在 ARCHIVELOG 模式下运行数据库，服务器参数文件和控制文件的恢复操作都是相同的。

12.6.1　故障点数据库恢复

对于故障点(point-of-failure)数据库恢复(也称为完全数据库恢复)来说，我们希望联机重做日志是完整无损的；实际上，任何未归档的联机重做日志都必须是完整无损的。如果丢失了联机重做日志，就必须执行数据库的不完全恢复操作。您可以参阅第 15 章以了解有关不完全恢复的更多内容。最后，我们还要假设至少存在一个完整的控制文件。如果不存在完整的控制文件，就需要恢复一个控制文件备份，如果联机重做日志不是完整的，就还需要执行数据库的不完全恢复操作。

在第一个示例中，我们丢失了所有数据库数据文件。而联机重做日志和控制文件完整无损，此时我们只需要恢复数据库。在这种情况下，可以选择执行完全恢复操作来恢复故障点的数据库。下面是执行这个恢复操作的命令示例：

```
shutdown;
startup mount;
restore database;
recover database;
alter database open;
```

下面是一个实际的还原操作示例：

```
RMAN> restore database;
Starting restore at 09-JAN-06
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=156 devtype=DISK
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
```

```
restoring datafile 00001 to
C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\SYSTEM01.DBF
restoring datafile 00002 to
C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\UNDOTBS01.DBF
restoring datafile 00003 to
C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\SYSAUX01.DBF
restoring datafile 00004 to C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\USERS01.DBF
restoring datafile 00005 to
C:\ORACLE\DATAFILE\TESTOEM\TESTOEM\CATALOG01.DBF
restoring datafile 00009 to C:\ORACLE\TESTDBF\TABLESPACE_NOASSM_01.DBF
restoring datafile 00010 to C:\ORACLE\TESTDBF\TABLESPACE_YESASSM_01.DBF
channel ORA_DISK_1: reading from backup piece
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\BACKUPSET\2006_01_08\O1_MF_NNNDF_TAG20060108T224324_1W3THND6_.BKP

channel ORA_DISK_1: restored backup piece 1
piece handle=C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\BACKUPSET\2006_01_08\O1_MF_NNNDF_TAG20060108T224324_1W3THND6_.BKP
tag=TAG20060108T224324

channel ORA_DISK_1: restore complete, elapsed time: 00:07:46
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00006 to C:\ORACLE\TESTDBF\TABLESPACE_2K.DBF

channel ORA_DISK_1: reading from backup piece
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\BACKUPSET\2006_01_08\O1_MF_NNNDF_TAG20060108T224324_1W3TTCT2_.BKP

channel ORA_DISK_1: restored backup piece 1
piece handle=C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM\
BACKUPSET\2006_01_08\O1_MF_NNNDF_TAG20060108T224324_1W3TTCT2_.BKP
tag=TAG20060108T224324
channel ORA_DISK_1: restore complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00007 to C:\ORACLE\TESTDBF\TABLESPACE_4K.DBF
channel ORA_DISK_1: reading from backup piece
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\BACKUPSET\2006_01_08\O1_MF_NNNDF_TAG20060108T224324_1W3TTNWO_.BKP

channel ORA_DISK_1: restored backup piece 1
piece handle=C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\BACKUPSET\2006_01_08\O1_MF_NNNDF_TAG20060108T224324_1W3TTNWO_.BKP
tag=TAG20060108T224324

channel ORA_DISK_1: restore complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting datafile backupset restore
```

```
channel ORA_DISK_1: specifying datafile(s) to restore from backup set

restoring datafile 00008 to C:\ORACLE\TESTDBF\TABLESPACE_16K.DBF
channel ORA_DISK_1: reading from backup piece
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\BACKUPSET\2006_01_08\O1_MF_NNNDF_TAG20060108T224324_1W3TTYJM_.BKP

channel ORA_DISK_1: restored backup piece 1
piece handle=C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\TESTOEM
\BACKUPSET\2006_01_08\O1_MF_NNNDF_TAG20060108T224324_1W3TTYJM_.BKP
tag=TAG20060108T224324
channel ORA_DISK_1: restore complete, elapsed time: 00:00:03
Finished restore at 09-JAN-06
RMAN> recover database;
Starting recover at 09-JAN-06
using channel ORA_DISK_1
starting media recovery
media recovery complete, elapsed time: 00:00:12
Finished recover at 09-JAN-06

RMAN> alter database open;
database opened
```

可以看出，上面介绍的操作非常简单。但对这样的还原操作还有几点需要注意。首先，如果某个文件已经存在并且不需要被恢复，Oracle 就不会恢复这个文件。阅读完 Oracle 文档后，您可能会认为：如果丢失了单个数据文件，只需要执行 restore database 命令，并且 Oracle 会只恢复丢失的数据文件。

而实际上，Oracle 会确定要还原的文件是否已经存在。如果已存在，并且已有的文件与准备还原的文件相同，RMAN 就不会再次还原这个文件。如果备份映像上的文件与已有的数据文件有所不同，RMAN 则会恢复这个文件。因此，如果丢失了一个或两个数据文件，您可能需要执行一个数据文件或表空间恢复操作，而不需要执行整个数据库恢复操作(数据文件恢复会更快)，我们将简要讨论这个问题。

现在先简单介绍一下还原/恢复进程中每个步骤所执行的操作。对于任何类型的 ARCHIVELOG 模式还原操作来说，这些步骤都是非常相似的。恢复了服务器参数文件和控制文件之后(如果有必要)，可以执行 restore database 命令使 RMAN 开始还原所有的数据库数据文件。需要注意的是，在这种情况下，由于我们要还原关键的表空间(即 SYSTEM 表空间)，所以必须关闭数据库。尽管可以联机执行许多 ARCHIVELOG 模式的还原操作，但是不能联机执行完全时间点恢复操作。

一旦还原了数据文件，Oracle 会执行下一条命令(即 recover database 命令)。这条命令与在 SQL*Plus 中执行的 recover database 命令非常相似，都是通过应用归档的重做日志(执行完全时间点恢复时要用的)使 Oracle RDBMS 开始将数据库恢复到故障点。此外，RMAN 的另一个优点是它可以从磁盘上还原所需的归档的重做日志，这样就可以在恢复进程中应用这些归档的重做日志。一旦 Oracle 恢复了数据库，就可以简单地执行 alter

database open 命令使 Oracle 完成打开数据库的进程。

第 11 章曾详细讨论了使用多信道能够加快备份速度的原因，对数据库恢复来说，使用多信道同样能够加快恢复速度。如果使用两个不同的信道将数据库备份到两个不同的设备上，则在还原操作执行期间，也应该分配两个信道(假设用于还原的硬件是相同的)。这样在从普通备份以及使用多个分区大小的备份还原时可以获得并行处理带来的好处。

注意：
如果尝试执行完全数据库还原操作失败，所有恢复的数据文件都会被删除。如果花费很长时间才完成了还原操作，那么最终导致的失败结果是很令人沮丧的。建议您测试不同的恢复策略(如一次恢复 4 个或 5 个表空间)，观察哪种策略的恢复操作最佳，哪种方法最能满足您的恢复 SLA 和灾难恢复需求。

RMAN 实习：完全恢复 ARCHIVELOG 模式的数据库

实习注意事项

在这个实习中，需要一个 Oracle 安装软件和一个可操作的 Oracle 测试数据库。

注意：
对于这个实习来说，数据库必须在 ARCHIVELOG 模式中配置和运行。

步骤 1 确保配置了控制文件的自动备份操作：

```
set oracle sid=recover
rman target rman_backup/password
configure controlfile autobackup on;
```

需要注意的是，在这种情况下，我们会在默认的位置中创建控制文件备份集片。

步骤 2 由于这是一个联机备份，所以不需要关闭和加载这个数据库。完成系统的联机备份。在这种情况下，我们会备份数据库和归档的重做日志。一旦备份了归档的重做日志，我们将删除这些归档的重做日志。在这个实习中，假定将在配置的默认设备上创建备份：

```
backup database plus archivelog delete input;
```

步骤 3 关闭数据库：

```
shutdown immediate;
```

步骤 4 重命名所有数据库数据文件和控制文件(如果没有足够的空间或出现了问题您可以承担得起丢失数据库的损失，也可以选择删除这些文件)。在这个实习中不要重命名联机重做日志。

步骤 5　启动和加载数据库，并且还原控制文件：

```
startup nomount;
set DBID = <enter the DBID of the database here>;
restore controlfile from autobackup;
alter database mount;
```

步骤 6　RMAN 使用步骤 2 中生成的备份来恢复这个数据库：

```
restore database;
recover database;
alter database open resetlogs;
```

步骤 7　通过再次备份数据库完成数据库的恢复操作：

```
shutdown immediate;
startup mount;
backup database;
```

12.6.2　表空间恢复

有时，我们可能只丢失了某个表空间的数据文件。此时，我们可以选择只恢复这个表空间，而不用恢复整个数据库。表空间恢复可以在不关闭数据库的情况下进行。例如，假设您丢失了自己的应付账表空间，但是您的应收账表空间仍然可用，只要应用程序不需要访问应付账表空间，您就可以在保持应收账表空间可访问的同时恢复应付账表空间。下面是恢复一个表空间需要使用的代码示例：

```
sql "alter tablespace users offline";
restore tablespace users;
recover tablespace users;
sql "alter tablespace users online";
```

您可能会发现这个恢复进程非常简单。首先，需要使要恢复的表空间脱机，使用 sql 命令可以完成这个操作。跟在 sql 命令后面的封装在双引号中的是我们希望数据库执行的 SQL；这里，我们执行 alter tablespace users offline 命令使 USERS 表空间脱机。接下来，还原与这个表空间关联的数据文件，然后恢复这个表空间。最后，再次使用 sql 命令来执行 alter tablespace users online 命令，并完成 USERS 表空间的恢复操作。

注意：
不能将单个表空间或数据文件还原至与数据库其余部分所处时间点不同的某个时间点。

如下所示，也可以在相同的命令组中恢复多个表空间：

```
sql "alter tablespace users offline";
sql "alter tablespace data offline";
restore tablespace users, data;
recover tablespace users, data;
sql "alter tablespace users online";
sql "alter tablespace data online";
```

12.6.3　数据文件恢复

数据文件恢复与表空间恢复非常相似，它是恢复数据库的基本途径。在这里，我们可以仅替换丢失的数据库数据文件，同时保持表空间其余部分的联机状态。数据文件恢复允许 DBA 恢复指定的数据文件，与此同时允许表空间的其余部分保持联机状态，这样用户仍然可以访问这些部分。如果数据文件相对于整个表空间来说是空的或极少被填充，这就是一个特别有用的功能。下面是恢复数据文件需要使用的代码示例：

```
sql "alter database datafile 3 offline ";
sql "alter database datafile 'd:\oracle\oradata\users01.dbf' offline ";
restore datafile 3;
restore datafile 'd:\oracle\oradata\users01.dbf';
recover datafile 3;
recover datafile 'd:\oracle\oradata\users01.dbf';
sql "alter database datafile 3 online";
sql "alter database datafile 'd:\oracle\oradata\users01.dbf' online ";
```

在这个示例中，使用了两种不同的方法定义了要恢复的数据文件，并且恢复了两个数据文件。首先，我们再次使用了 sql 命令和 alter database datafile offline 命令使受损的数据文件脱机 (在某些时候受损的数据文件可能已经脱机，这里是为了确保这些数据文件脱机)。然后使用 restore datafile 命令还原数据文件。第一个命令使用编号还原数据文件。第二个命令使用文件名还原数据文件。

下一步将恢复数据文件。同样，采用通过编号还原第 1 个数据文件，然后根据文件名恢复数据文件的方法。最后使用 SQL 命令使数据文件回到联机状态。同样在第 1 个 SQL 命令中使用数据文件编号，在第 2 个 SQL 命令中使用数据文件名。

在执行下一条命令前，先介绍一下如何在 alter database 命令中引用数据文件。有两种不同的方法可以引用数据文件：第一种方法是通过号码引用数据文件，在示例中我们引用了数据文件 3；第二种方法是通过文件名引用数据文件，在示例中我们引用了数据文件'd:\oracle\oradata\ users01.dbf'。我们可以使用上述的任何一种方法，不过使用数据文件号码比较容易。通常在数据文件丢失或出现问题时，Oracle 会在相关的错误消息中同时给出数据文件的名称和号码，下面是一个错误消息的示例：

```
ORA-01157: cannot identify/lock data file 4 - see DBWR trace file
ORA-01110: data file 4: 'D:\ORACLE\ORADATA\RECOVER\TOOLS01.DBF'
```

注意在这条错误消息中，数据文件 4 与 tools01.dbf 数据文件关联。通常，指示还原数据文件 4 比指示还原 d:\oracle\oradata\recover\tools01.dbf 简单得多。

一旦使数据文件脱机，就可以还原和恢复这些数据文件(再次使用数据文件号码或数据文件名)。最后，我们再次使数据文件联机，并完成恢复进程。

12.6.4　使用增量备份的恢复操作

Oracle 会在还原数据文件时自动判断是否使用了增量备份策略，还会根据需要自动应用所需的增量备份集。在这些情况下，不需要执行不同的恢复操作。

在使用增量备份的还原操作期间，restore 命令只还原基本备份。一旦完成了这个还原操作，就可以执行 recover 命令将增量备份和归档的重做日志应用于数据库。一旦完成了恢复操作，我们就能够照常打开数据库。在所有情况下，Oracle 都会尝试还原基本备份和最新的增量备份。这会减少完全恢复数据库所应用的重做，从而减少还原时间。

需要注意的是，由于在恢复进程期间数据库很可能会应用多个备份集，所以恢复操作的时间会超出我们的预期。不过，根据多种因素(数据更改速率是一个重要因素)，应用增量备份集比应用大量重做日志更快，这样增量备份解决方案就是一个快速的解决方案。因此，增量备份的优势在于它是一个快速的备份策略(并且需要更小的空间来保存备份集片)，不过它的代价是恢复时间有可能会更长。

12.7　联机重做日志丢失的恢复

用户可能经历的一类恢复情况是丢失了数据库联机重做日志。联机重做日志的丢失包含以下四种不同的情况：

- 重做日志文件组成员丢失
- 非活动的联机重做日志组丢失
- 活动但非当前的联机重做日志组丢失
- 当前联机重做日志组丢失

前两种类型最多是令人烦恼的事情，但后两种类型对数据丢失来说简直就是灾难。前面章节中讨论过当系统执行提交命令或联机重做日志缓冲区被占用一定空间后，将导致重做日志的写入(当然也可能有其他事件导致重做日志的写入)。结果是，尚未提交的 undo 与提交的更改都被写入联机重做日志中。由于数据库数据文件的写入具有延迟性，因此数据库数据文件中的数据通常与数据库当前数据的实际状态不能保持同步。这一情况导致 Oracle 在数据库恢复时需要应用联机重做日志的重做。

由于数据库数据文件与实际数据库的状态不能保持同步，活动的联机重做日志的丢失将直接导致数据的丢失。当前联机重做日志的丢失也会导致数据的丢失。显然，重做日志十分重要。上述提到的三种重做日志状态：当前的，活动的，非活动的重做日志主要差别如下：

- **活动的(Active)**：该类联机重做日志并非当前正在使用的重做日志。但是它包含尚未写入到数据文件中的重做。其日志组可能(也可能不)需要被归档。
- **当前的(Current)**：该类联机重做日志为当前的联机重做日志组。Oracle 正在将重做写入到该联机重做日志组中。
- **非活动的(Inactive)**：该类联机重做日志不是当前正在使用的联机重做日志组。其重做已经被 DBWR 进程写入到数据文件中。

可以通过查询 V$LOG 视图的 STATUS 列查看联机重做日志组的状态。以下内容将讨论在丢失重做日志组的情况下需要执行的操作。

12.7.1　非活动的联机重做日志文件组成员的丢失

对联机重做日志组丢失一个或几个成员(不是整个组都丢失)的情况进行恢复相对比较容易。可以通过使用 alter database add logfile member 命令重建成员。丢失重做日志成员的情况可以通过 OEM 的警告或在警告日志中获得：

```
ORA-00313: open failed for members of log group 2 of thread 1
ORA-00312: online log 2 thread 1 : 'C:\ORACLE\ORADATA\DANCE\REDO01a.LOG'
```

建议使用的最佳实践是，如果数据库尚未关闭并且丢失的成员是活动重做日志组或当前重做日志组的成员，则立刻使用 alter system checkpoint 命令进行检查点检查。该命令将强迫数据库以紧迫方式将所有数据库缓冲区缓存的脏数据块写入到数据库数据文件中。这有助于保护联机重做日志的数据免于丢失而导致数据库崩溃。

一旦完成检查点检查操作，则应该执行 alter database add logfile 命令重建重做日志组成员 redo02a.log：

```
SQL>Alter database add logfile 'C:\ORACLE\ORADATA\DANCE\REDO02a.LOG'
    reuse to group 2;
```

若在添加日志文件前数据库已崩溃，则应该加载数据库，然后执行 alter database add logfile 命令。此后才能够打开数据库。

另外一个可用的选项是以一致性方式关闭数据库(使用 shutdown normal 命令)，将重做日志组的另外一个成员复制到丢失的成员中。然后以正常方式重启数据库。

注意：
强烈建议在丢失联机重做日志组或该组成员时不要使用 shutdowm abort 命令关闭数据库。

12.7.2　非活动的联机重做日志文件组的丢失

丢失非活动状态的联机重做日志文件组相对比较容易恢复。需要了解可能发生的两种不同情况。第一种情况是在数据库启动期间丢失非活动状态的联机重做日志文件组；第二种情况是在数据库操作期间丢失非活动状态的联机重做日志文件组。下面的内容将讨论如何处理这两种不同的情况。

1. 在数据库启动期间丢失非活动的联机重做日志文件组

在启动数据库时，若不能打开非活动状态的联机重做日志文件组，将看到如下的错误消息：

```
ORA-00313: open failed for members of log group 2 of thread 1
ORA-00312: online log 2 thread 1 : 'C:\ORACLE\ORADATA\DANCE\REDO02a.LOG'
```

首先要做的工作是确定联机重做日志组中是否有成员存在，如果有成员存在，按照

前述的处理丢失联机重做日志文件组成员的步骤执行恢复操作。如果所有成员都丢失，则使用如下所示的 alter database 命令将该日志文件组删除：

```
SQL>alter database drop logfile group 2;
```

一旦删除联机重做日志文件组后，只需要使用 alter database add logfile 命令重建联机重做日志文件组：

```
SQL>Alter database add logfile
    1 group 2 'C:\oracle\oradata\dance\redo02a.log'size 50m;
```

2. 在数据库运行期间丢失非活动的联机重做日志文件组

在数据库运行期间丢失非活动的联机重做日志文件组(或者毁坏)，此时数据库仍然可以继续操作。Oracle 数据库有时会跳过丢失的联机重做日志文件组继续正常运行。此时，执行 alter system checkpoint 命令，然后使用 alter database clear logfile group 命令清除、日志文件组，如下所示：

```
SQL>alter system checkpoint;
SQL>alter database clear logfile group 1;
```

若需要清除的日志文件尚未归档，将会出现下面的错误：

```
SQL> alter database clear logfile group 1;
alter database clear logfile group 1
*
ERROR at line 1:
ORA-00350: log 1 of instance orcl (thread 1) needs to be archived
ORA-00312: online log 1 thread 1: '/oracle01/oradata/dance/redo01.log'
```

由于日志文件并不存在，当然不能对其进行归档。在此情况下，使用 alter database clear unarchived logfile 命令清除未归档的日志文件并在其当前位置重建日志文件，如下所示：

```
SQL>alter database clear unarchived logfile   '/oracle01/oradata/dance/
redo01.log';
```

由于归档重做日志将丢失，因此需要立即备份数据库。

有时，尽管数据库没有崩溃，但却处于冻结状态。此时，打开另外一个 SQL*Plus 会话并连接到数据库。然后执行 alter database checkpoint 命令，然后根据恢复需要的类型，执行 alter database clear logfile 命令或 alter database clear unarchived logfile 命令。在执行完上述命令后，数据库将恢复正常操作。

12.7.3　丢失活动但非当前状态的联机重做日志文件组

如果遭遇丢失活动的联机重做日志文件组的情况，则需要使用前节示例所使用的 alter database clear unarchived logfile 命令。因为活动的联机重做日志未被归档。需要指示

Oracle 这种情况是正常的。该命令将重建联机重做日志并允许 Oracle 继续执行正常的操作。在执行该操作后应该经常备份数据库。

12.7.4 丢失当前联机重做日志文件组

没有比丢失当前联机重做日志文件组更糟糕的事情了。在此情况下，一般都会丢失数据。如果丢失当前联机重做日志文件组，一般数据库都会关闭而不会处于正常状态。

幸运的话，数据库可能没有关闭，此时应该立即使用 alter system checkpoint 命令执行检查点检查数据库操作，以避免将数据全部丢失，然后快速将数据库关闭。使用 alter system checkpoint 命令将强制数据库将数据库缓冲区缓存中的脏数据块写入到数据库数据文件中。

不需要任何的恢复可能也能打开数据库。按照如下方式重新启动数据库：

(1) 执行 startup mount 命令；

(2) 执行 alter database clear unarchived logfile 命令重建丢失的重做日志组。本章前面的内容中对该命令有介绍。

(3) 执行 alter database open 命令。

如果幸运的话，数据库将被成功打开。

如果数据库未被打开，您将处于糟糕的境地。需要执行数据库不完全恢复。本书第 14 章将详细介绍使用 RMAN 执行不完全恢复操作。

12.8 数据恢复顾问

Oracle Database 11*g* 提供了一个新的功能—自动诊断存储库(Automatic Diagnostic Repository，ADR)。ADR 提供了大量新的信息以及新的工具以方便数据库管理，本节将讨论其中一个新的工具，数据恢复顾问。

数据恢复顾问有 OEM 接口，同时也提供命令行接口。本节主要涉及相关的命令行接口，如果希望使用 OEM 接口，则可以参考第 13 章的有关使用 OEM 和 RMAN 的内容。

通过 RMAN 使用数据恢复顾问

利用 RMAN 命令行接口，就可以使用手动接口进入数据恢复顾问。ORACLE 提供了一些新的 RMAN 命令以实现从命令行进入数据恢复顾问。这些命令包括：

- list failure 命令
- advise failure 命令
- repair failure 命令
- change failure 命令

所有数据恢复顾问(DRA)命令都需要在数据库实例启动的情况下工作。如果产生新的错误且数据库关闭时，在将数据库打开并检测到错误以前，通常不会得到任何结果。一旦检测到错误，DRA 将保存该错误，可以使用 advice failure 命令在数据库处于未加载、加载或打开阶段时访问 DRA 中的有关错误。

在修正错误时数据库应处于哪种状态取决于错误的类型。例如要恢复控制文件，数据库需要处于未加载状态。若需要修复丢失的数据文件，则执行 repair failure 命令使数据库处于加载或打开状态。

我们在实践中也发现，在一些情况下，repair failure 命令不能正确完成任务。最常见的情况是在恢复控制文件时。通常，执行 repair failure 命令后，往往要执行 recover database 命令，然后使用 alter database open resetlogs 命令以实现完全还原。当然，DRA 可以帮助初级的 DBA 着手处理存在问题的数据库。

在处理数据崩溃错误时，一般使用的流程是首先使用 list failure 命令，然后使用 advice failure 命令，最后执行 repair failure 命令。下面将详细讨论如何使用这些命令。

list failure 命令　RMAN 的 list 命令包含新的 failure 参数，使用该参数可列出其检测出的错误以及错误的优先级别(关键、高、低)，状态(打开、关闭)，以及对错误的汇总。在此环境下，一个错误就是指当前存在于系统中的持久的数据损坏。以下是一个使用 list failure 命令的示例：

```
RMAN> list failure;
using target database control file instead of recovery catalog
List of Database Failures
=========================
Failure ID Priority Status    Time Detected Summary
---------- -------- --------- ------------- -------
187        HIGH     OPEN      05-JUN-09     One or more non-system
                                            datafiles are missing
```

注意上例中的输出，丢失的数据文件并不会被列出。可以使用 list failure detail 命令生成关于该错误的其他细节。另外，list failure exclude failure *n* 命令能够使您从报告输出中排除特定的错误号。其他的选项包括只列出处于关闭状态的错误，只列出关键错误，只列出具有高优先级或低优先级的错误，根据错误 ID 列出或排除错误。以下为使用相关选项的示例：

```
RMAN> list failure detail;
List of Database Failures
=========================
Failure ID Priority Status    Time Detected Summary
---------- -------- --------- ------------- -------
187        HIGH     OPEN      09-JUN-09     One or more non-system
                                            datafiles are missing
List of child failures for parent failure ID 187
Failure ID Priority Status    Time Detected Summary
---------- -------- --------- ------------- -------
2075       HIGH     OPEN      09-JUN-09     Datafile 6:
'C:\ORACLE\ORADATA\ROB11GR4\ROB11GR4\BOOGLE01.DBF' is missing
  Impact: Some objects in tablespace BOOGLE might be unavailable

-- Let's exclude failure_id 187…
```

```
RMAN> list failure exclude failure 187;
no failures found that match specification
```

表 12-2 给出了使用 list failure 命令所有可用的选项的列表。

表 12-2 list failure 命令可用选项

选 项	描 述
ALL	列出所有错误
CRITICAL	只列出优先级为关键的错误
HIGH	只列出优先级为高的错误
LOW	只列出优先级为低的错误
CLOSED	只列出已经关闭的错误
EXCLUDE FAILURE	从列表中，去除列出错误号的错误。该选项后跟随一个由逗号隔开的希望去除的错误 ID 列表
DETAIL	提供列出的错误的其他细节
#	实际列出的错误号

以下是其他示例(注意，为节省空间，一些不重要的输出被忽略):

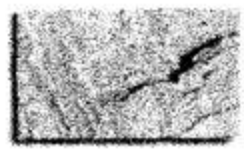
```
RMAN> list failure closed 254;
List of Database Failures
=========================
Failure ID Priority Status    Time Detected Summary
---------- -------- --------- ------------- -------
254        HIGH     CLOSED    04-JUN-09     Redo log file
C:\ORACLE\ORADATA\DOODLE\REDO03.LOG is corrupt

RMAN> list failure closed exclude failure 242,248;
List of Database Failures
=========================
Failure ID Priority Status    Time Detected Summary
---------- -------- --------- ------------- -------
8298       CRITICAL CLOSED    03-AUG-09     System datafile 1:
'C:\ORACLE\ORADATA\DOODLE\SYSTEM01.DBF' needs media recovery
8295       CRITICAL CLOSED    03-AUG-09     Control file needs media
                                            recovery
8104       CRITICAL CLOSED    03-AUG-09     Control file
C:\ORACLE\ORADATA\DOODLE\CONTROL03.CTL is missing
8101       CRITICAL CLOSED    03-AUG-09     Control file
C:\ORACLE\ORADATA\DOODLE\CONTROL02.CTL is missing
8098       CRITICAL CLOSED    03-AUG-09     Control file
C:\ORACLE\ORADATA\DOODLE\CONTROL01.CTL is missing
```

注意：
list failure 命令只能在单一实例的数据库中运行(因此，RAC 集群必须处于单实例模式)。也不能在物理备用数据库上使用该命令。

list failure 命令本身并不能检查数据库错误。数据库一直不断地在检查损坏问题，这些问题周期性地被记录到数据字典中(以及物理 ADR 存储库中，该存储库在磁盘上而不是在数据库中)。

在数据库关闭时，如果有错误发生，在需要使用丢失的组件时，才能检测到该错误。例如，如果控制文件丢失，只有在数据库加载时才会检测到该错误。如果数据文件丢失，在执行打开数据库操作时才能检测到该错误。如果在数据库打开时需要使用该数据文件，则错误将会被检测到而数据库也被打开。

如果出现在列表中的一个错误处于打开状态，则意味着对当前的问题需要着手处理。通过查看新的 advise failure 命令，将会列出与该问题关联的一个或多个修复操作。这些选项有助于在处理该问题时确定使用何种修复选定来修正错误。下面的内容将讨论该命令。

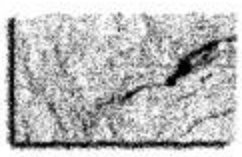


注意：
如果一个数据文件脱机，则该数据文件不会作为错误被报告。如果该数据文件物理丢失，则将会作为错误报告。

advise failure 命令　当 list failure 命令列出了一个处于打开状态的错误，advise failure 命令可用于提供可使用的改正该错误的建议方案。以下是使用 advise failure 命令的示例：

```
RMAN> advise failure;
List of Database Failures
=========================
Failure ID Priority Status    Time Detected Summary
---------- -------- --------- ------------- -------
187        HIGH     OPEN      05-JUN-09     One or more non-system
                                            datafiles are missing
analyzing automatic repair options; this may take some time
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=129 device type=DISK
analyzing automatic repair options complete

Manual Checklist
================
1. If file C:\ORACLE\ORADATA\ROB11GR4\ROB11GR4\DOOGLE01.DBF was
unintentionally renamed or moved, restore it

Automated Repair Options
========================
Option Repair Description
------ ------------------
1      Restore and recover datafile 6
```

```
  Strategy: The repair includes complete media recovery with
            no data loss
Repair script:
C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\rob11gr4\hm\reco_1214740950.hm
```



注意：
与 list failure 命令一样，advise failure 命令也只能在单一实例的数据库上远行(因此，RAC 集群必须置于单实例模式)。该命令也不能在物理备用数据库上使用。

通过上面的示例，应该注意到从提供的输出上，RMAN 提供手动和自动修复错误的选项。自动修复选项包含了可用于修复该错误的 RMAN 命令。基于不同的数据库状态(例如加载或未加载)，这些自动修复选项可能不同。建议尽可能打开数据库。例如，如果可以成功地将数据库加载，就应该尽量将数据库加载而不是让数据库处于未加载模式。

另外还应注意修复选项可以包括数据丢失，如果使用给定的恢复选项，数据恢复顾问将指出是否会发生数据丢失现象。这些命令被包含在 ADR 结构内的一个文件中。以下是恢复文件的示例：

```
# restore and recover datafile
    restore datafile 6;
    recover datafile 6;
    sql 'alter database datafile 6 online';
```

可以选择手动运行恢复文件，也可以使用 repair failure 命令，以下将讨论该问题。

repair failure 命令 在检测到错误并根据 Oracle 的建议确定了恢复方案后，则可以手动修复错误，也可以通过使用 repair failure 命令允许 Oracle 自动修复错误。为运行 repair failure 命令，必须启动目标数据库实例。如果需要修复多个错误，Oracle 将为它们建立统一的修复操作方案。RMAN 也将核查存在的错误，如果仍然有错误尚未改正就不能执行恢复操作。以下是一个从 RMAN 执行 repair failure 命令的示例(为简单起见，将部分 RMAN 输出省略)：

```
RMAN> repair failure;
    Strategy: The repair includes complete media recovery with no data loss
    Repair script:
    C:\ORACLE\PRODUCT\diag\rdbms\rob11gr4\rob11gr4\hm\reco_110341808.hm
    contents of repair script:
      # restore and recover datafile
      restore datafile 6;
      recover datafile 6;
      sql 'alter database datafile 6 online';
    Do you really want to execute the above repair (enter YES or NO)? yes
    executing repair script
    Starting restore at 05-JUN-09
    using channel ORA_DISK_1
    ... Typical RMAN restore output is removed for brevity...
```

```
Starting recover at 05-JUN-00
using channel ORA_DISK_1
... Typical RMAN recover output is removed for brevity...
media recovery complete, elapsed time: 00:00:03
Finished recover at 05-JUN-09
sql statement: alter database datafile 6 online
repair failure complete
```

注意：
repair failure 命令也只能应用于运行单一实例的数据库(因此，RAC 集群必须置于单实例模式)。注意该命令不能修复 RAC 集群中的指定节点无法访问的诸如数据文件等类似错误。

如果需要查阅修复错误的操作，则可以使用 repair failure preview 命令。该命令将会显示要应用的修复活动，但不会执行修复操作。

change failure 命令　RMAN 的 change 命令包含一个新的 failure 关键字，利用该关键字可以修改由 Oracle 数据库检测到的错误状态。例如，可以修改指定错误的优先级，或将所有错误从高优先级改为低优先级。也可以选择关闭一个或多个错误。默认情况下，RMAN 将给出提示以确认希望作出的更改。可以使用 change 命令中的 noprompt 子句强制执行修改而无需提示。以下是一个将错误 187 的优先级改变为 LOW 的示例：

```
RMAN> Change failure 187 priority low;
List of Database Failures
=========================
Failure ID Priority Status    Time Detected Summary
---------- -------- --------- ------------- -------
187        HIGH     OPEN      09-JUN-09     One or more non-system
                                            datafiles are missing
Do you really want to change the above failures (enter YES or NO)? yes
changed 1 failures to LOW priority
```

注意：
不能将状态为 CLOSED 的错误切换到 OPEN 状态。

数据恢复顾问数据字典视图

为支持数据恢复顾问，Oracle 11g 增加了一些新的视图。这些视图以 V$IR 为前缀，如表 12-3 所示。

表 12-3　以 V$IR 为前缀视图

视 图 名 称	描　　述
V$IR_FAILURE	提供关于错误的信息。注意该视图中的记录可能在视图中存在父记录
V$IR_FAILURE_SET	该表提供了各种与错误关联的指导记录的列表，允许使用 V$IR_FAILURE 视图与 V$IR_MANUAL_CHECKLIST 视图连接
V$IR_MANUAL_CHECKLIST	该视图提供了与错误有关的详细的信息消息。这些消息提供了如何手动修复错误问题的信息
V$IR_REPAIR	当该视图与 V$IR_FAILURE 和 V$IR_FAILURE_SET 连接时，提供了由 Oracle 建立的，用于改正检测到的错误的，包含修复步骤的物理文件的指针

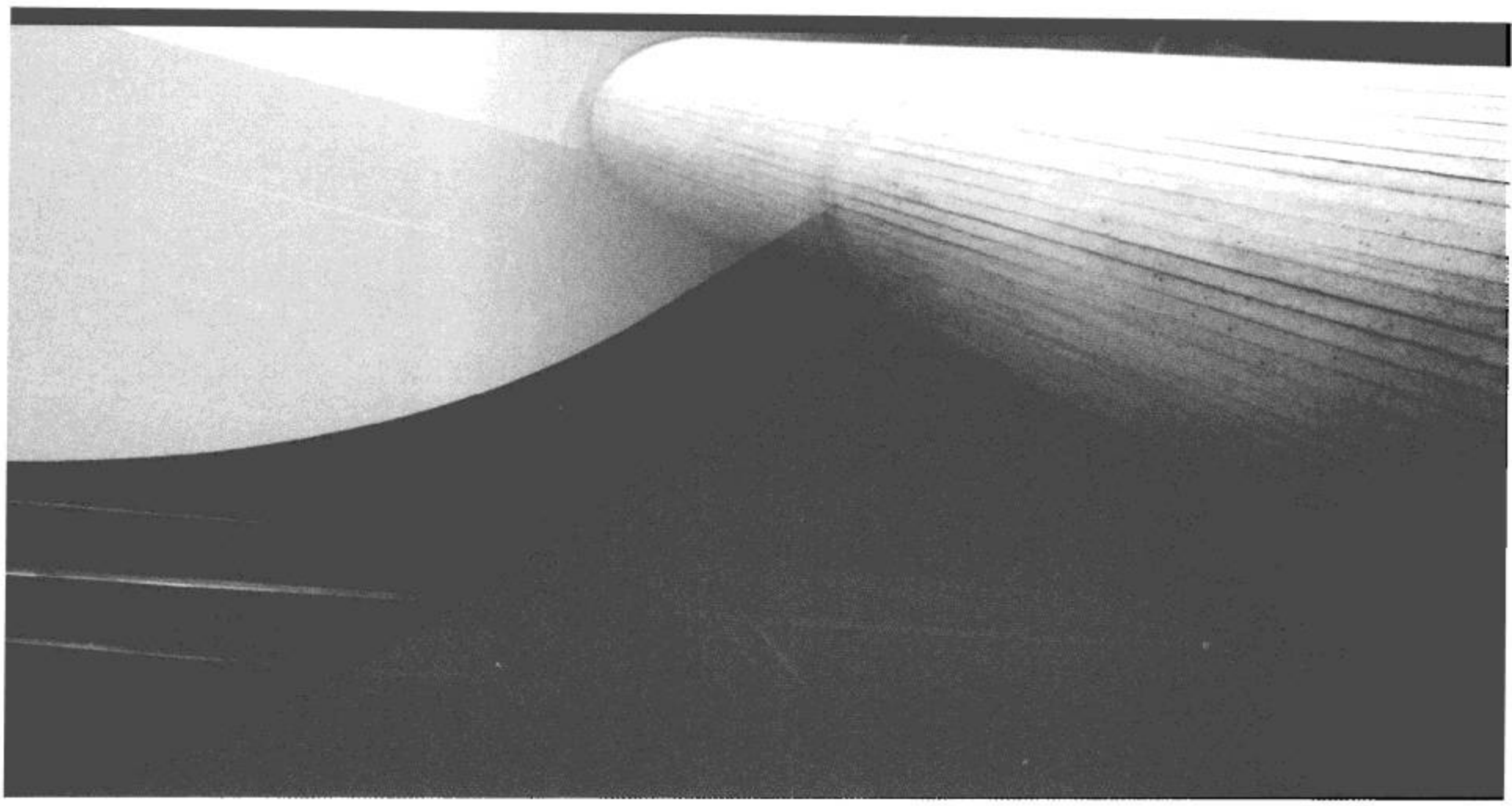
以下是对 DRA 视图查询的一个示例：

```
-- Do we have an open error reported?
select failure_id, time_detected, description from v$ir_failure
Where status='OPEN';

FAILURE_ID TIME_DETE DESCRIPTION
---------- --------- ------------------------------------------
       242 19-SEP-07 One or more non-system datafiles are missing
       605 19-SEP-07 Datafile 4: '/oracle01/oradata/orcl/user
                     s01.dbf' is missing
```

12.9　小结

本章详细讲述了 RMAN 的基本恢复操作。在介绍完 RMAN 还原与恢复的基础知识之后，我们讲述了如何以不同方式还原服务器参数文件和控制文件。之后介绍的是如何使用 restore 和 recover 命令从 RMAN 备份中还原和恢复数据库。然后介绍了不同的恢复选项，从完整数据库恢复到特定表空间或数据文件的恢复。最后，我们提供了一些实习来帮助实践学到的恢复技能。

最后，我们建议要不断地实践恢复操作，从而了解 RMAN 的工作原理并可以实际地恢复数据库而不需要翻书。试着成为您工作领域的 RMAN 专家吧，您终将会成为拯救系统的英雄！



第Ⅲ部分

更有效地使用 RMAN

第 13 章　使用 Oracle 企业管理器实现备份和恢复操作

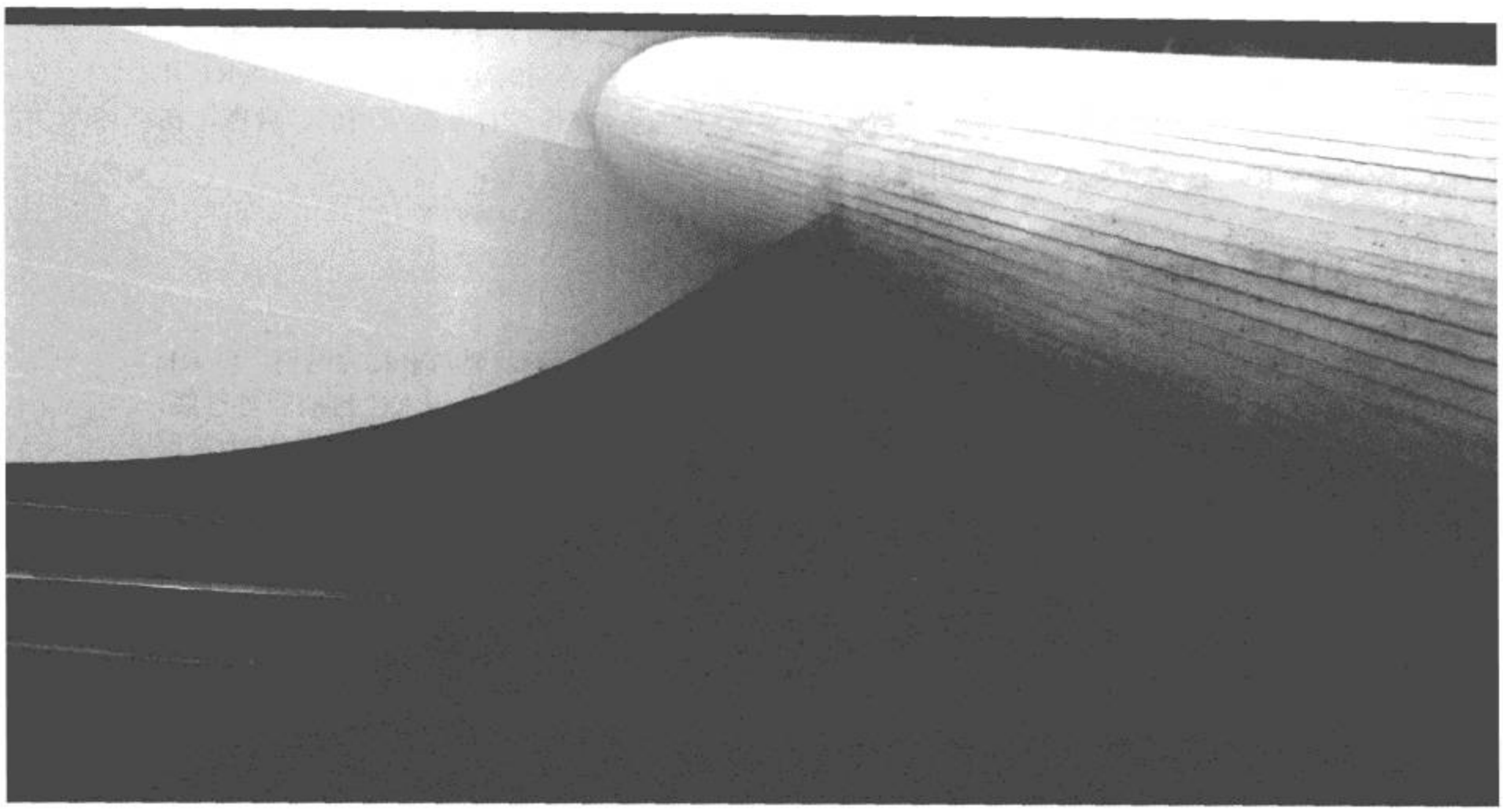
第 14 章　RMAN 的高级恢复主题

第 15 章　从用户错误中恢复：闪回技术

第 16 章　RMAN 维护

第 17 章　RMAN 的监控和报告

第 18 章　调整 RMAN 备份与恢复操作的性能



第 13 章

使用 Oracle 企业管理器实现备份和恢复操作

迄今为止，我们已经提供了用 RMAN 客户端实用程序与 RMAN 交互的指导。这使我们对使用 RMAN 命令行语法建立了一些信心。学会使用这种语法非常重要，因为在一些情况下，RMAN 的命令行语法是在令人痛苦的停机期间唯一可用的工具。

Oracle 提供了工具集来监控业务涉及的所有数据库，并且该工具集包括一个用于获得备份和执行恢复的图形用户界面。该产品称为 Oracle 企业管理器(Oracle Enterprise Manager，OEM)，从 1998 年诞生以来它一直以这样或那样的形式存在。在大多数时候，该产品一直用作可以安装在客户端系统上的 Java 应用程序，并且用于监控和管理远程数据库。

然而，从 10g 开始，企业管理器(Enterprise Manager，简写为 EM)经历了彻底的变革，它从桌面应用程序变成了一个全能的 Web 应用程序。从早期的 10g Release 1(10gR1)开始，新的 EM 就开始不断更新，这使其几乎所有的功能在命令行都是可用的。现在，在任何 Internet 浏览器上也都是可用的。

13.1　Oracle 企业管理器：新范例

Oracle 专门重新设计了 OEM 以包括企业计算的新范例，通常被称为网格。Oracle 构建 OEM 来控制网格计算所引入的管理混乱。网格计算支持许多引人注目的业务进展，并且提供不断发展的、高冗余的、基于常用工具的、面向服务的环境，在该环境中可以部署企业计算资源。我们在前面已经介绍了部署在客户站点上的网格计算的原理，它的功能非常强大。但是也非常复杂。

下面通过查看一些示例环境来比较传统企业计算和网格计算之间的区别。首先，考虑传统企业环境的简化情况，如图 13-1 所示。

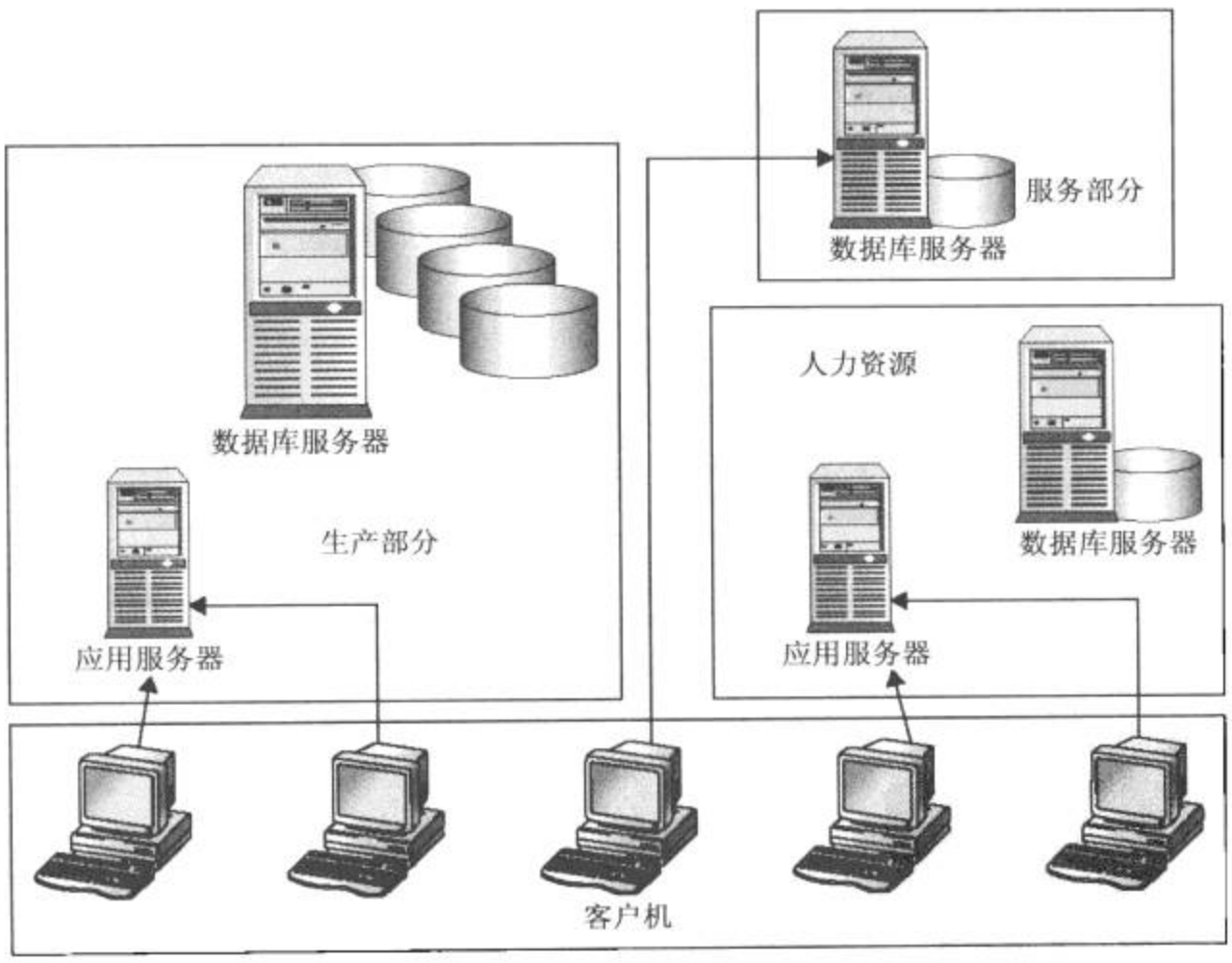


图 13-1　传统企业计算

配置这个数据库环境需要如下步骤：

- 请求可以找到的、功能尽可能强大的计算机。
- 管理可以分组的、尽可能小的计算机列表。

- 在一台计算机管理多个数据库。
- 按照不同的部分和功能分离数据。
- 将一组正式的责任分配给每个专业程序段(网络管理、系统管理、数据库管理等)。

如果业务需求相同并且将在网格计算环境中部署它们，如图 13-2 所示，则配置环境的步骤如下：

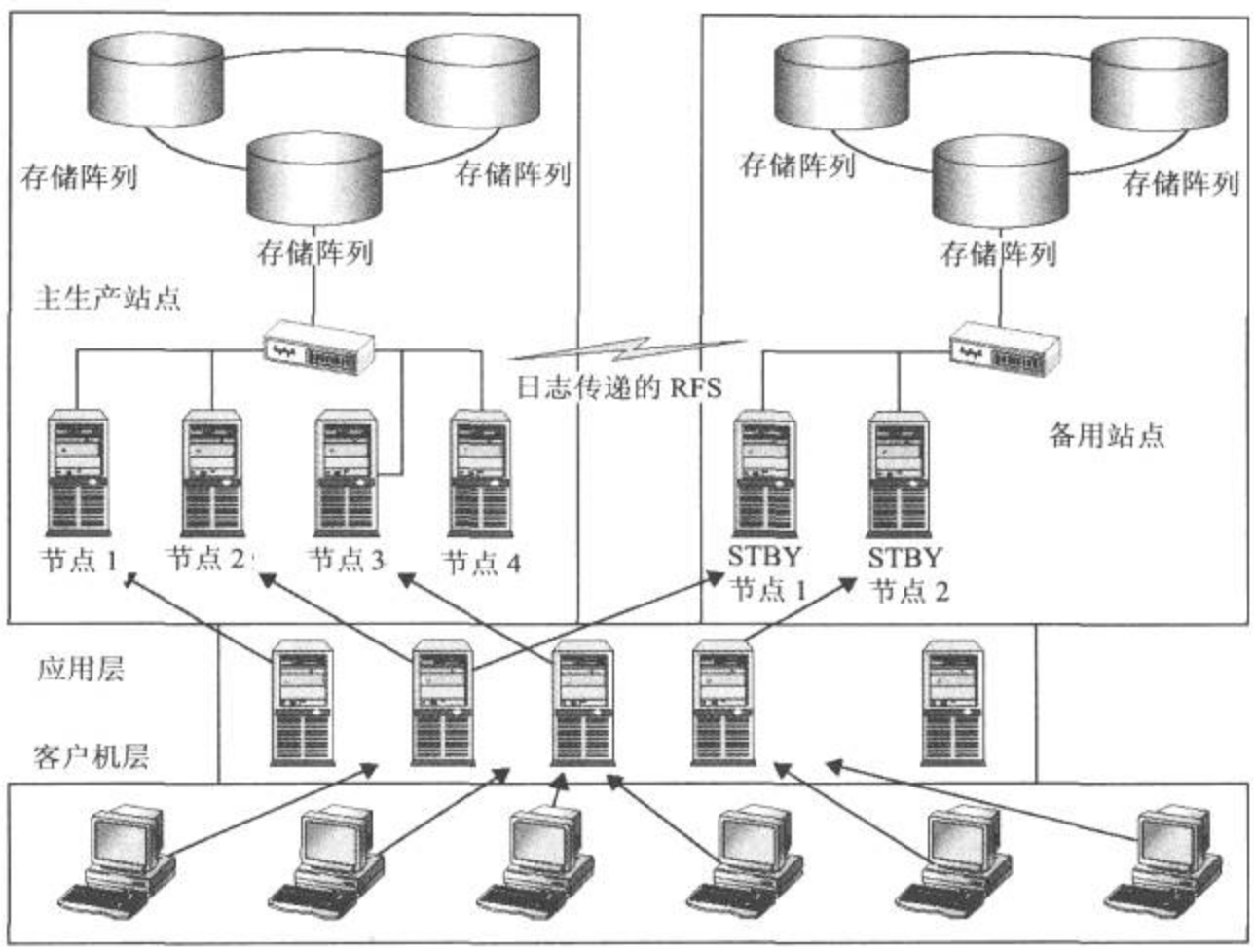


图 13-2　企业网格

- 请求可以找到的、尽可能多的常用计算机。
- 管理运行在多台计算机上的一个数据库。
- 作为抽象化(服务)分离数据。
- 快速承担新的专业责任。
- 作为一个资源管理企业计算，根据使用需求将企业计算分布到各个部分和功能。

这种不需要停机就可以增加或减少资源的网格能力会大大地增加其复杂度。DBA 会发现自己每天需要做更多的体系结构操作工作、网络管理工作，以及系统管理工作。

OEM 10*g* 的设计用来处理网格计算引入的新问题。OEM 10*g* 根据面向服务体系结构的原理构建，它的功能远比 Oracle 9*i* 企业管理器强大。现在，OEM 能够监控和管理整个“Oracle 生态系统”：主机服务器、磁盘存储、数据库、应用服务器和应用程序。

涵盖所有 OEM 的功能超出了这本书的范围(为您推荐 *Oracle Enterprise Manager Grid Control 10g Handbook*)。本书主要介绍 RMAN，因此对 OEM 的介绍只限于它如何通过控制台来为 RMAN 提供备份和恢复界面。然而，在讨论备份和恢复功能之前，了解体系结构和 EM 的整体功能对于学习 OEM 是有益处的。

OEM 10g 的形式有两种：Grid Control(网格控制)和 Database Control(数据库控制)。OEM 10g 的 Grid Control 是具有完整功能的全企业 Oracle 生态系统管理工具。OEM 11g 的 Database Control 是可以只作为数据库管理实用程序部署的 OEM 版本。

Grid Control 和 Database Control 之间的区别

Enterprise Manager(企业管理器)的 Grid Control(网格控制)可以监控整个 Oracle 生态系统。Grid Control 具有一个中心存储库，用于收集有关多个计算机上的多个目标的数据，并且提供一个界面来显示所有已发现目标的共同信息。

Enterprise Manager(企业管理器)的 Database Control(数据库控制)是 Grid Control 功能的一个子集。Database Control 只监控一个数据库，并且不能用于监控多个数据库。它只在数据库上运行。

从数据库管理的观点来看，两个实用程序的功能几乎是相同的，只不过网格控制提供了涉及多台计算机多个数据库的操作的更多功能。但是，两个实用程序的界面是相同的，底层代码也是相同的，并且只有很细微的区别。从备份和恢复的观点来看，两个实用程序几乎相等——唯一主要的区别就是在生态系统间共享 RMAN 脚本的能力和利用 Grid Control 多个服务器的知识来复制和修复的能力。

13.2 Grid Control

首先需要了解的是，OEM 是一个 Web 应用程序，它具有 Web 应用程序自带的所有功能和限制。OEM 控制台是运行在 HTTP 服务器上的 Web 页面，该 HTTP 服务器将作为 Grid Control 安装的一部分安装和配置。没有任何客户端安装。

其次，Grid Control 部署在 Oracle 应用服务器(Oracle Application Server，OAS)上。当安装 Grid Control 时，同时也会安装 OAS，然后 Grid Control 应用程序作为 Oracle Containers for J2EE(OC4J)应用程序部署在 OAS 之上。指出这一点是为了使您了解：提交 Grid Control 的用户需要在某种程度上熟悉 OAS 体系结构。对于 OAS 的讨论超出了本书的范围，但我们会在本书中经常引用 OAS。实际上，对 Grid Control 本身的完整讨论也超出了本书的范围。Grid Control 可以监控许多不同类型的目标(当 Grid Control 调用它们时)：数据库、应用服务器、主机自身，甚至是存储设备。在本书中，我们介绍数据库的监控，并且在大多数情况下只监控数据库的备份。

Grid Control 执行的任务说起来很简单，但实际去了解却很复杂：它收集遍及企业的计算系统的相关信息，将这些信息合并到中心存储库中，然后通过它的 Web 控制台向 DBA 显示这些信息。根据这些信息，DBA 可以让 Grid Control 代表 DBA 在这些计算系统中执行任务。

Oracle Client 安装光盘上的 OEM 客户端安装简介

仍然存在可以作为独立产品安装的“瘦”Java 客户端。这是较早的 Oracle 9*i* 企业管

理器客户端软件，可以在 Oracle Client 安装光盘上找到该软件。这被继续保留在 10g 中，因为某些功能还没有从桌面应用程序转变为 Database Control(数据库控制，Web 应用程序。在 11g 中，数据库操作的所有的功能已经被集成到 Web 应用程序，而且不再是桌面可安装的 OEM。

13.2.1　Grid Control 的体系结构

Grid Control 的体系结构(如图 13-3 所示)从 Oracle 管理服务(Oracle Management Service，OMS)开始，这是部署在应用服务器上的应用程序。OMS 通过中心代理程序(central agent)从注册的目标服务器中收集数据。中心代理程序安装在目标服务器上，它收集信息并且将数据推入 OMS。OMS 将数据加载到存储库数据库中，然后基于存储库中的信息构建 Web 页面，可以通过能够定位 OMS 服务器 URL 的任意浏览器检索该 Web 页面。

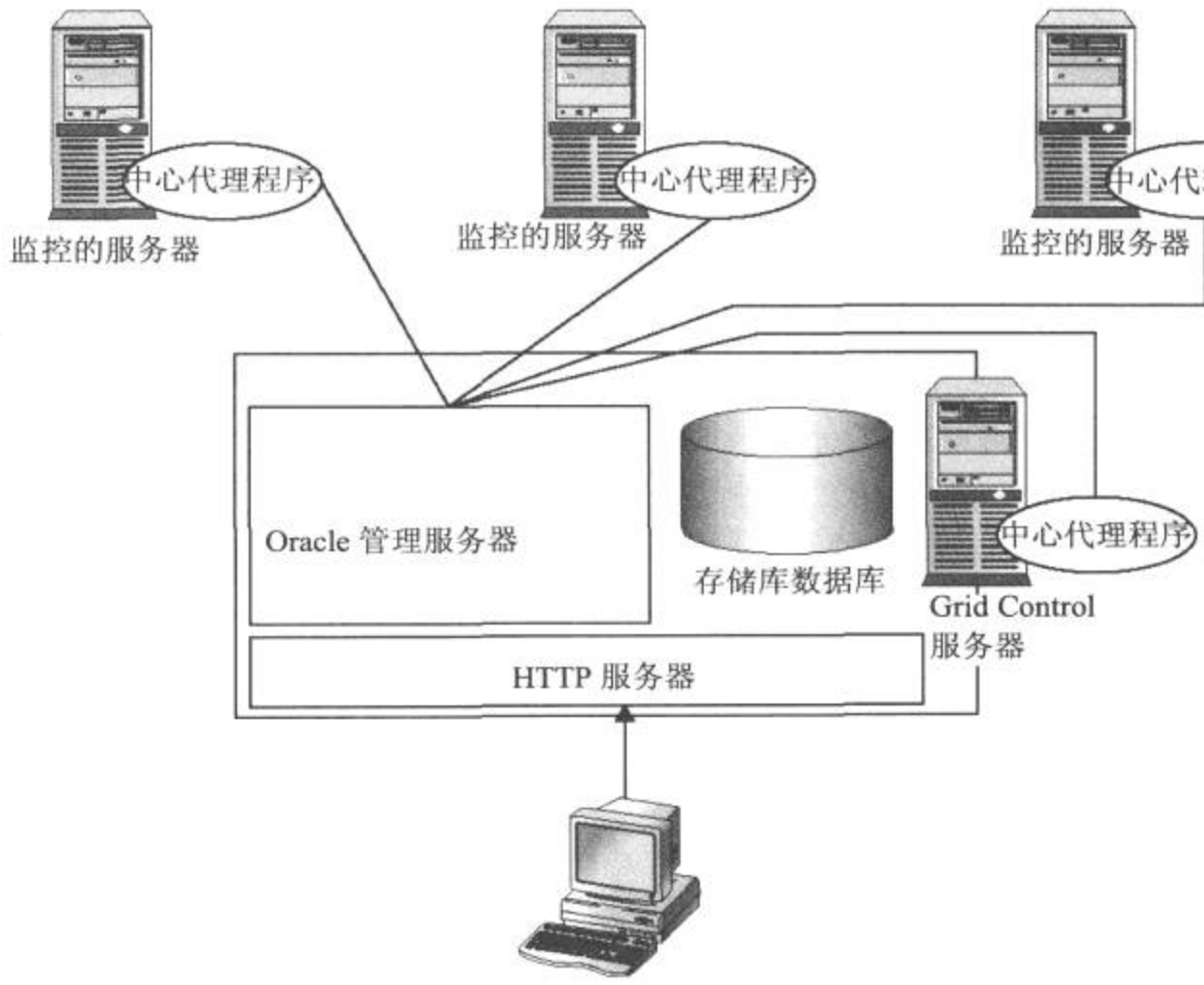


图 13-3　Grid Control 体系结构

1. 中心代理程序

中心代理程序安装在使用 Grid Control 监控的每台计算机上。中心代理程序是“哑”软件片，即它不可以独立做出任何决策。中心代理程序使用 Perl 脚本收集数据，将这些数据通过 HTTPS 推入 OMS。OMS 执行必需的任何智能操作，然后将操作发送给代理程序以执行该操作。

从存储的观点来看，中心代理程序占用相对较少的存储空间。然而，取决于其所执行的操作，中心代理程序可能占用大量的 CPU 资源。

2. Oracle 管理服务(OMS)

OMS 是在 Grid Control 体系结构的中间层上部署的 Web 应用程序，它不断从代理程序处以.xml 文件的形式接收信息，然后将这些信息加载到存储库的表中。它负责为提供控制台 Web 页面的 HTTP 服务器构建 Web 内容，由此可能需要中心代理程序提供特定的信息。然而，数据一般是从中心代理程序推入的，而不是由 OMS 推入。

3. 存储库数据库

OMS 使用 Oracle 数据库作为它的数据源。存储库数据库用于存储有关管理目标以及 Grid Control 操作(例如作业或通知)的信息。

在存储库中可以很好地利用高级的 RDBMS 特性；可以使用高列队(Advanced Queuing，AQ)功能，大量使用分区技术，甚至是使用内部的 DBMS_JOB。

13.2.2 安装和配置 Grid Control

在以前的版本中，本书用了几页的内容来介绍 Grid Control 的安装。由于其他的书已经把 Grid Control 安装的内容介绍的很好了，而且本书将给出与利用企业管理器的 Database Control 有关的例子，这里我们就不介绍 Grid Control 的安装了。Oracle 出版社出版的 *Oracle Enterprise Manager Grid Control 10g Handbook* 这本书详细介绍了 Grid Control。尽管如此，关于 Grid Control 还要做最后一点说明，Grid Control 不是随意在什么地方都可以安装的。当测试安装的时候要非常小心，因为很容易出现资源过载(特别是在 RMAN 书中提到的那种极低的测环境下)。

注意：
如果在同一个服务器上同时运行 Grid Control 中心代理程序和 Database Control，那么必须不遗余力的让自己知道每个步骤中应该在哪个$ORACLE_HOME 目录中执行操作，这是因为 emctl 可执行文件存在于每个 ORCLE_HOME 目录(包括驻留代理程序的 ORACLE_HOME 目录)的 bin 目录中，但如果开始在错误的 ORACLE_HOME 目录中运行了错误的 emctl 命令，就会产生严重的问题。对此，我们建议在执行任何操作之前运行一组环境更改脚本。

13.3 Database Control

前面介绍过，Database Control 是 Grid Control 功能的一个子集，它只能管理和监控至多一个数据库。因此，Database Control 的功能是以数据库为中心，但又不完全限于数据库。由于会有一些为数据库所在的服务器收集的主机统计信息，因此主机(并非单独的

“目标”)有一些报告的信息。此外，可以配置 Database Control 以监控和管理自动存储管理(ASM)实例，目标数据库可以使用该 ASM 实例进行存储。

也可以配置 Database Control 以监控 RAC 数据库，由此监控多个主机、多个实例和多个 ASM 实例。当然，它只可以监控一个数据库——因此 Database Control 的限制仍然是相同的。

13.3.1　Database Control 的体系结构

Database Control 的体系结构非常类似于 Grid Control 的体系结构，但规模小很多。中心代理程序和 OMS 合并到相同的 OC4J 应用程序，并且存储库驻留在目标数据库本身中。图 13-4 显示了这种体系结构。

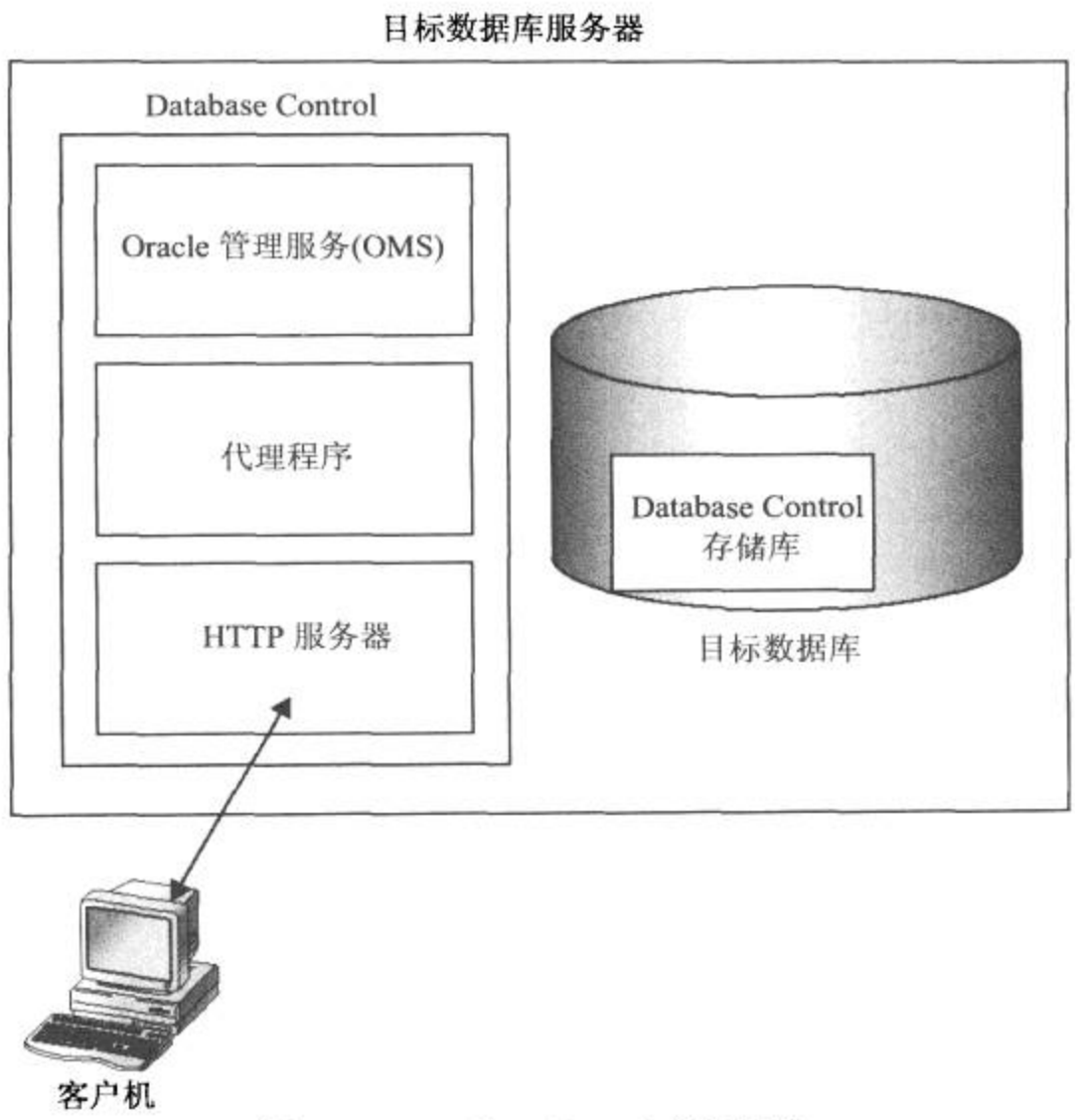


图 13-4　Database Control 体系结构

图 13-4 中非常重要的一点是，Database Control 的存储库位于它监控的目标数据库中。因此，如果目标数据库停机，Database Control 就不能正常运作，除非启动和停止该数据库，并且执行恢复。在 Windows 系统上，Database Control 进程是单独的进程，并且作为 Services 控制面板中单独的服务启动和停止。在 Linux 和 Unix 系统上，dbconsole(数据库控制台)作为单独的应用程序被停止和启动，但它仍然会产生单独的代理程序和 Java 进程。

Database Control 与 RDBMS 一起安装，并且它所在的 ORACLE_HOME 目录与其目

标数据库的相同。特别地，在为数据库配置 dconsole 后，就会在 ORACLE_HOME 中创建名为 *host_sid* 的新子目录，其中 *host* 是计算机名，而 *sid* 则是所监控的实例的 SID。例如，如果服务器名为 horatio.hadba.com，并且监控的数据库名为 v112，则配置文件均位于$ORACLE_HOMEhoratio.haba.com_v112 中。

在 RAC 环境中，一般根据群集中的每个节点命名多个目录，后面跟上 RAC 数据库的 globalname(全局名)。因此，如果有两个节点的 RAC 群集，两个节点分别为 dex 和 horatio，并且数据库为 v112c，则在每个节点的 RDBMS 主目录上有两个目录：ORACLE_HOME/dex_v112c 和$ORACLE_HOME/horatio_v112c。在每个节点上，只有名称与节点对应的目录才会实际地拥有文件。而另一个目录则为空。这种约定在 RAC 群集共享同一个位于共享驱动器上的$ORACLE_HOME 软件树时也适用。

在目标数据库的$ORACLE_HOME/bin 目录中停止和启动 Database Control 的命令很简单，如下所示：

```
emctl start dbconsole
emctl stop dbconsole
```

13.3.2 安装和配置 Database Control

Database Control 软件不需要进行任何特定的安装。Database Control 默认与已有的每个 10.2 版本的(或者更高版本) RDBMS 一起安装。实际上，如果具有在安装时提供启动数据库(starter database)的通用安装程序，就可能已经在计算机上启动和运行 Database Control。此时，安装程序会要求您选择管理数据库的方式，其中一个选项就是 Database Control。

1. 使用数据库配置助手配置 Database Control

Oracle 提供了一个 GUI 实用程序来帮助构建数据库，该 GUI 实用程序一般称为数据库配置助手(Database Configuration Assistant，DBCA)。由于数据库配置助手由位于$ORACLE_HOME/bin 目录外部的可执行程序 dbca 启动，因此，我们在下文中均使用小写字母 dbca 表示数据库配置助手。

dbca 的用途很广——构建新的数据库，删除原有的数据库，安装数据库选项等。也可以使用 dbca 向数据库添加 Database Control 功能，或者在构建新数据库时向其添加 Database Control，或者修改已有的数据库。这是到目前为止启动和运行 Database Control 最简单和最直观的方法。

dbca 可执行程序不可以用于删除或修改已经添加的 Database Control。它只是一个用于添加 Database Control 的工具；如果要删除 Database Control，则需要使用命令行工具 emca(下面将介绍该工具)。

图 13-5 给出了是否自动调度备份的选项，即 Enable Daily Disk Backup to Recovery Area 复选框。该选项启用 Oracle 建议备份策略，本章 13.7.1 小节将介绍该备份策略。这是一个即装即用的备份策略，也可以使用 emca 进行配置。

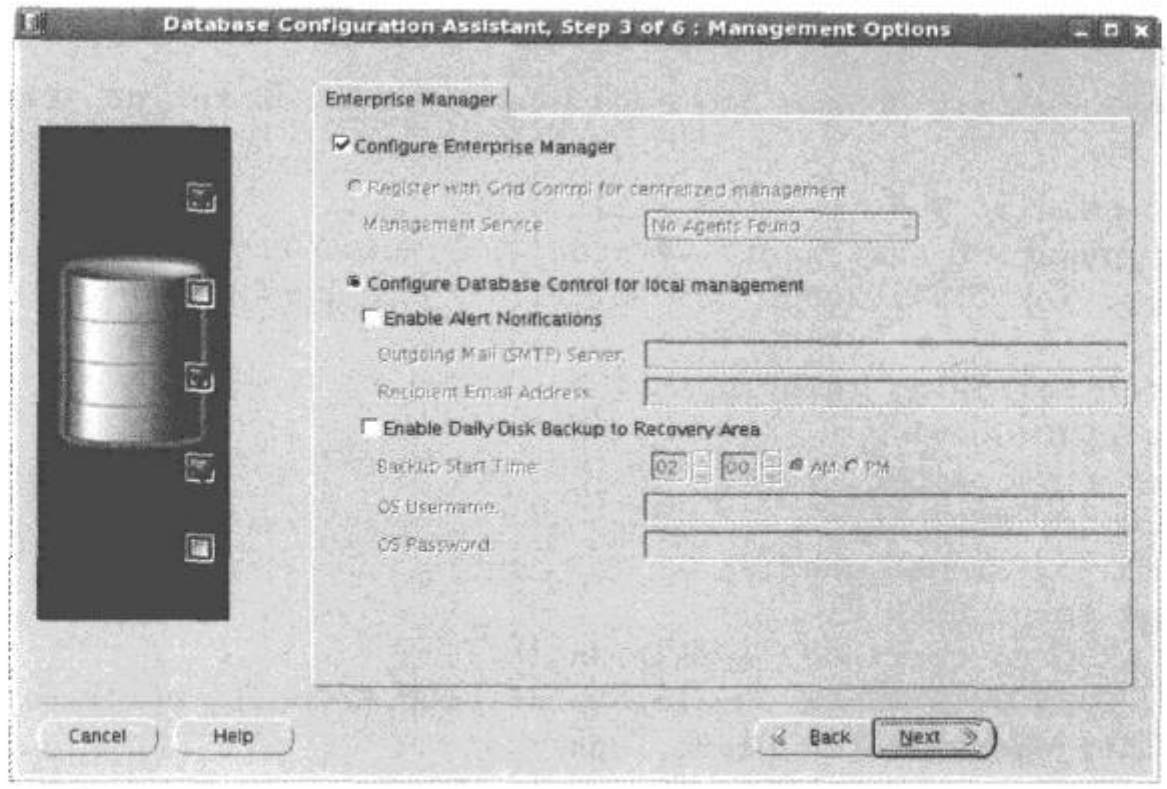


图 13-5　配置选项

2. 使用企业管理器配置助手配置 Database Control

和所有 Oracle 工具一样，GUI 有时会显得过于迟钝、死板或者难以理解，而无法用来配置 Database Control。在这些情况下，可以使用称为企业管理器配置助手(Enterprise Manager Configuration Assistant，EMCA)的命令行实用程序，可在 RDBMS 安装的 $ORACLE_HOME/bin 目录中使用 emca 可执行程序运行该实用程序(同样，我们约定用小写字母 emca 表示该实用程序)。

不同于 dbca，emca 的用法不是很直观。但是，它提供了各种调节按钮和开关，从而比 GUI 的 dbca 具有更强大的功能，且更易于调试。此外，最好避免使用不必要的 X 窗口。因此，emca 是值得花费一些时间学习的有价值的工具。Oracle 文档中全面地介绍了所有的开关和控件；从我们的角度来看，只有一些开关值得关注，即用于标识将要配置 Database Control 的数据库的开关，以及是否需要使用默认备份策略的开关：

```
emca -config dbcontrol db -repos create -backup
emca -deconfig dbcontrol db -repos drop
```

执行一条命令后，emca 以一种交互式的模式给出提示信息如下：配置(或取消配置)的数据库 SID、连接 emca 与数据库的监听器的监听端口，以及所需的用户口令。需要注意的是，如果使用-backup 开关，emca 将提示备份配置设置。

RMAN 实习：使用 emca 配置 Database Control

实习注意事项

该实习假定存在在 Linux 上运行的名为 v102 的可操作 10.2.0.1.0 数据库，并且详细介绍了如何使用命令行实用程序 emca 为该数据库配置默认的 Database Control 应用程序。

步骤 1 使用 emca 删除已有的 dbcontrol 配置和 dbcontrol 存储库:

```
[oracle@localhost dbhome_1]$ emca -deconfig dbcontrol db -repos drop

STARTED EMCA at Jun 11, 2009 3:03:30 PM
EM Configuration Assistant, Version 11.2.0.0.2 Production
Copyright (c) 2003, 2005, Oracle. All rights reserved.

Enter the following information:
Database SID: dev1
Listener port number: 1521
Password for SYS user:
Password for SYSMAN user:
Password for SYSMAN user:
Do you wish to continue? [yes(Y)/no(N)]: Y
Jun 11, 2009 3:03:52 PM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at
/home/oracle/app/oracle/cfgtoollogs/emca/dev1/emca_2009_06_11_15_03_30.log.
Jun 11, 2009 3:03:52 PM oracle.sysman.emcp.EMDBPreConfig performDeconfiguration
WARNING:EM is not configured for this database.No EM-specific actions can
be performed.
Jun 11, 2009 3:03:52 PM oracle.sysman.emcp.ParamsManager
checkListenerStatusForDBControl
WARNING:Error initislizing SQL connection.SQL operations cannot be performed
Jun 11, 2009 3:03:52 PM oracle.sysman.emcp.EMReposConfig invoke
INFO: Dropping the EM repository (this may take a while) ...

Jun 11, 2009 3:09:23 PM oracle.sysman.emcp.EMReposConfig invoke
INFO: Repository successfully dropped
Enterprise Manager configuration completed successfully
FINISHED EMCA at Jun 11, 2009 3:09:23 PM
```

步骤 2 使用 emca 为该数据库创建新的 dbcontrol 配置(注意下列输出中间部分有删减情况，emca 执行的步骤比以下要更多):

```
$ emca -config dbcontrol db -repos create
STARTED EMCA at Jun 11, 2009 3:16:23 PM
EM Configuration Assistant, Version 11.2.0.0.2 Production
Copyright (c) 2003, 2005, Oracle. All rights reserved.
Enter the following information:
Database SID: dev1
Listener port number: 1521
Listener ORACLE_HOME[/home/oracle/app/oracle/product/11.2.0/dbhom-1]:
Password for SYS user:
Password for DBSNMP user:
Password for SYSMAN user:
Email address for notifications (optional):
```

```
Outgoing Mail (SMTP) server for notifications (optional):
------------------------------------------------------------------
You have specified the following settings
Database ORACLE_HOME ................
/home/oracle/app/oracle/product/11.2.0/dbhome_1
Local hostname ................ horatio.hadba.com
Listener ORACLE_HOME ................
/home/oracle/app/oracle/product/11.2.0/dbhome_1
Listener port number ................ 1521
Database SID ................ dev1
Email address for notifications ..............
Outgoing Mail (SMTP) server for notifications ..............
------------------------------------------------------------------
Do you wish to continue? [yes(Y)/no(N)]: Y
Jun 11, 2009 3:16:58 PM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at
/home/oracle/app/oracle/cfgtoollogs/emca/dev1/emca_2009_06_11_15_16_23.log.
Jun 11, 2009 3:17:00 PM oracle.sysman.emcp.EMReposConfig createRepository
INFO: Creating the EM repository (this may take a while) ...
Jun 11, 2009 3:24:54 PM oracle.sysman.emcp.EMReposConfig invoke
INFO: Repository successfully created
...
Jun 11, 2009 3:37:09 PM oracle.sysman.emcp.EMDBPostConfig
performConfiguration
INFO: Database Control started successfully
Jun 11, 2009 3:37:09 PM oracle.sysman.emcp.EMDBPostConfig
performConfiguration
INFO: >\>\>\>\>\>\>\> The Database Control URL is
http://horatio.hadba.com:1158/em <<<<<<<<<<<

Jun 11, 2009 3:45:27 PM oracle.sysman.emcp.EMDBPostConfig invoke
WARNING:
************************** WARNING **************************
Management Repository has been placed in secure mode wherein Enterprise
Manager
data will be encrypted. The encryption key has been placed in the file:
/home/oracle/app/oracle/product/11.2.0/dbhome_1/localhost.localdomain_dev1
/sysman/
config/emkey.ora. Please ensure this file is backed up as the encrypted data
will become unusable if this file is lost.
************************************************************
Enterprise Manager configuration completed successfully
FINISHED EMCA at Jun 11, 2009 3:45:27 PM
```

13.4　在企业管理器中配置备份设置

虽然花费了一些时间来了解企业管理器，但是现在我们可以使用最近配置的 OEM Grid

或 Database Control 来获得数据库备份。我们希望您已经了解了 OEM 可以完成的工作，并且已经准备好开始安排备份计划，从而可以恢复每天的工作。在开始这部分之前，请记住我们使用的是 EM 的 Database Control，而不是 Grid Control。在此我们不指定某一个数据库——假设在控制台中只有一个数据库是可用的。

首先，需要配置备份设置。这意味着要在 OEM 控制台中单击一些页面。选择希望配置的适当数据库后，就可以在 Availability 选项卡中进行所有与备份和恢复相关的操作。

如图 13-6 所示，在 Availability 选项卡中，需要做的第一件事情是配置备份设置，因此单击名为 Backup Settings 的链接。Backup Settings 页面有 3 个选项卡：Device、Backup Set 和 Policy，下面的小节中将描述这 3 个选项卡。

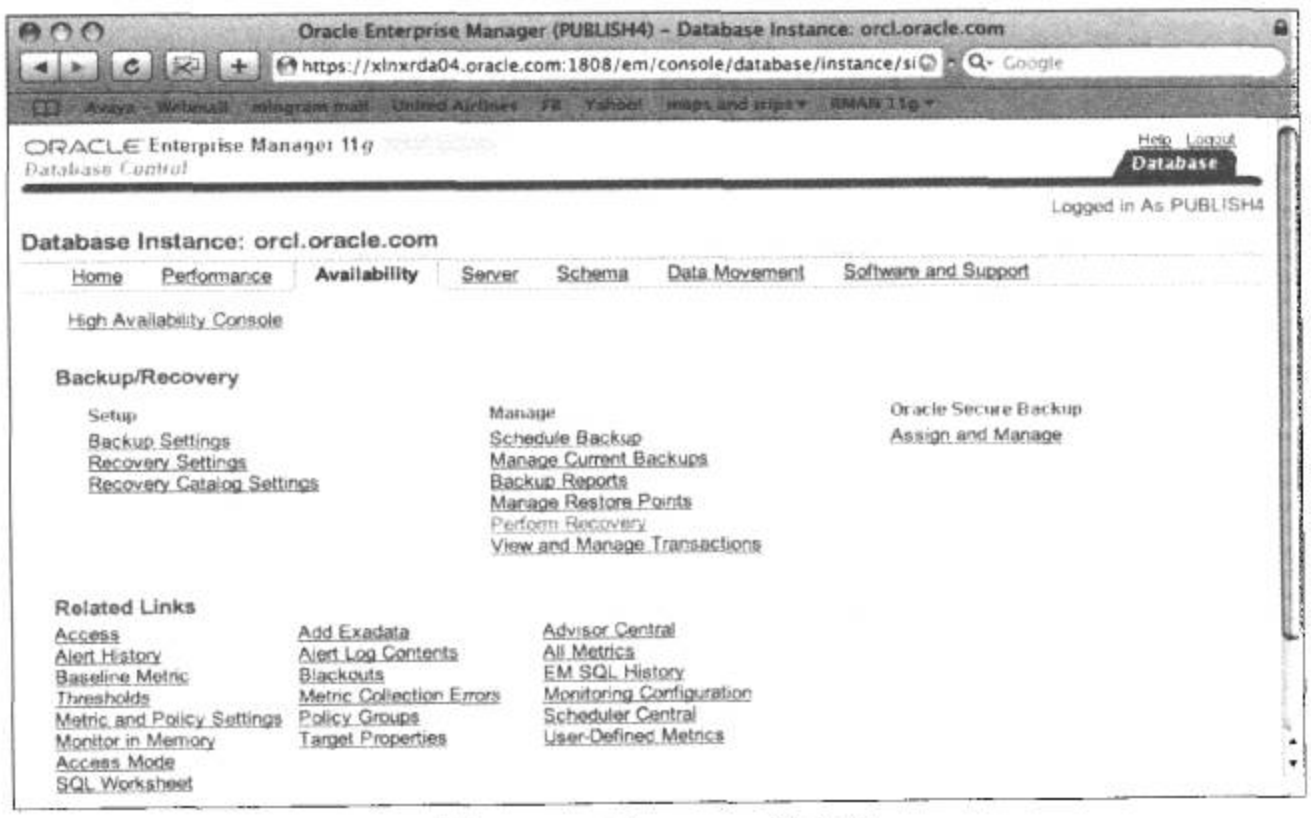


图 13-6 Maintenance 选项卡

13.4.1 设备配置

如图 13-7 所示，在 Device 选项卡中，可以设置磁盘和磁带设置。这些设置不是设置默认设备，而是针对这些设备上的所有信道的个别设置。对于磁盘备份，需要设置并行程度、备份位置以及备份类型(备份集或映像副本)。根据需要，也可以在此处开启压缩功能。对于磁带备份，需要告诉 OEM 使用多少磁带设备，以及磁带备份类型(压缩的或非压缩的)。此外，对于磁带来说，可以提供磁带备份软件运行所需的任何环境设置(关于磁带备份设置的更多内容，请参见第 4 章～第 8 章)。

需要注意的是，Device 选项卡的底部显示的是主机凭证(屏幕截图上未显示)。OEM 需要用这些主机凭证来提交作业，从而实现目标数据库上的更改。在 Configure Backup Settings 页面的每个选项卡的底部都应该显示主机凭证。

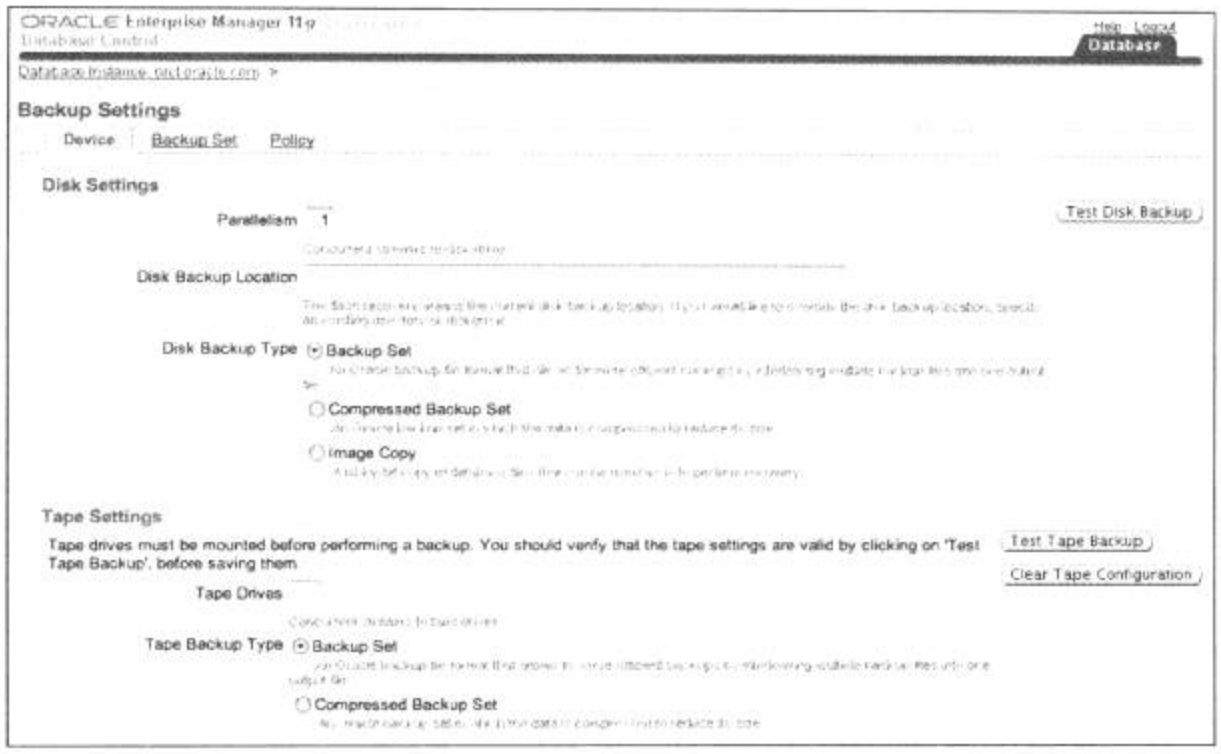


图 13-7　Device 选项卡

13.4.2　备份集配置

决定设备配置后，可单击 Backup Set 选项卡(如图 13-8 所示)，该选项卡是针对如何生成备份集进行的永久配置。记住，这只适用于使用备份集的备份，而不适用于使用映像副本的备份(磁盘备份使用备份集或映像副本，而磁带备份则总是使用备份集)。

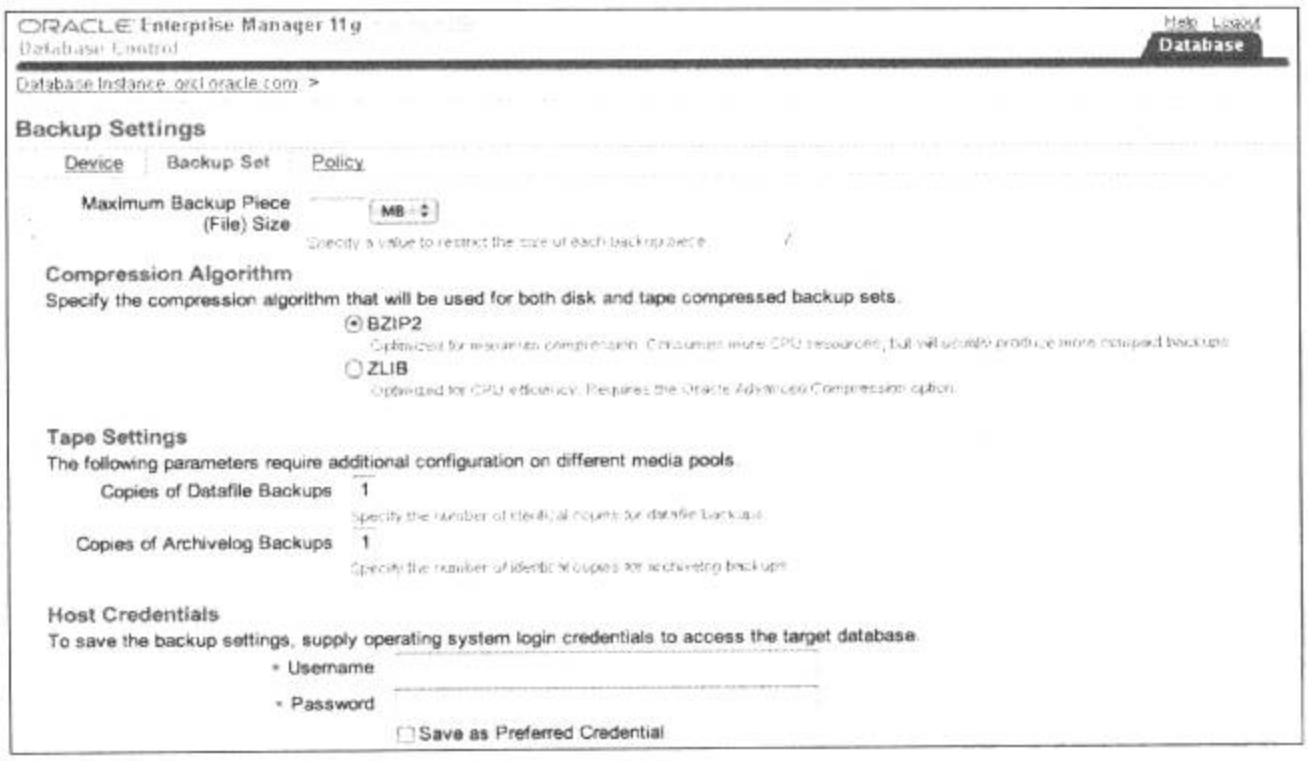


图 13-8　Backup Set 选项卡

如果使用磁盘备份，则只需要配置两个设置：备份片的最大尺寸和压缩算法。压缩可以在 BZIP2 和 ZLIB 之间选择——前者是对最大压缩的优化(以 CPU 周期为代价)，而后者是对低 CPU 利用率的优化(以更少的压缩为代价)。如果是备份到磁带，则可以设置在磁带设备上创建的每个数据文件备份集的副本个数，以及创建的每个归档日志备份的副本数量。

13.4.3 策略配置

Policy 选项卡用于配置与自己的业务备份策略相关的设置，包括打开控制文件和服务器参数文件(SPFILE)的自动备份，以及指定在何处放置自动备份(如果使用磁盘备份的话)。在 Policy 选项卡(如图 13-9 所示)上，可以打开备份优化(参见第 3 章以了解备份优化的详情)。在该选项卡上可以打开块更改跟踪(参见第 16 章)并且配置表空间排斥策略(参见第 3 章)。

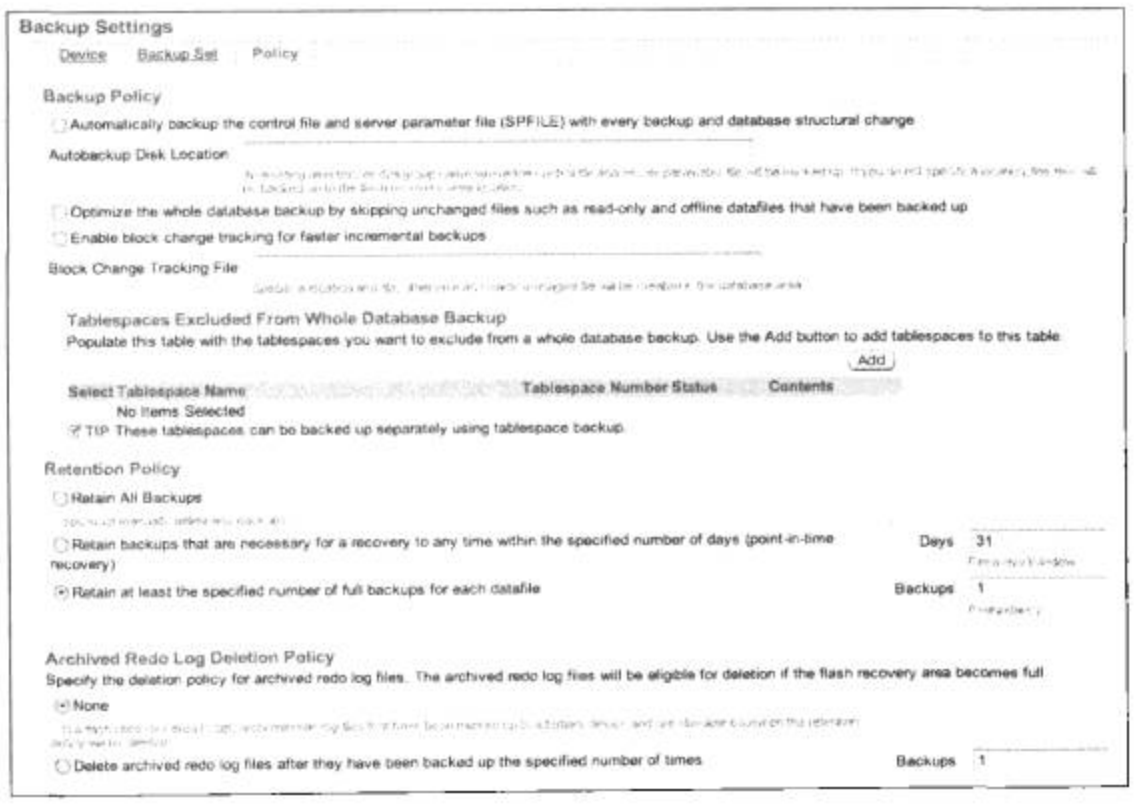


图 13-9 Policy 选项卡

也可以使用 Policy 选项卡配置保留策略。OEM 提供了 3 个选项："保存所有备份"、"根据恢复窗口设置策略"、或者"根据冗余设置策略"。如果选择恢复窗口或冗余，就需要分别指定天数或副本数。在此之后，再次提供主机凭证并单击 OK 按钮以提交改动。OEM 连接到目标数据库主机，并且执行 RMAN configure 命令以执行这些改动。

13.4.4 在 OEM 备份配置中无法配置的内容

在 OEM 中无法配置 RMAN 可以配置的所有内容。有一些改动只能通过 RMAN 命令行来完成：

- **备份加密**：当调度特定的备份时，加密启用，而不是在整体的备份设置中。当调度一个 Oracle 建议或自定义备份时，您可以指定加密级别。
- **默认设备类型**：在 OEM 中调度重复的备份，因此不需要默认设备类型。OEM 接口中不存在该类型。
- **归档删除策略**：在 RMAN 中，可以设置不同于备份集保留策略的特定归档日志保留策略。在 OEM 中没有这样的选项。
- **快照控制文件位置**：RMAN 允许修改快照控制文件位置，该功能对于 RAC 配置非常方便。OEM 没有方法实现这一点。

- **备份调速**：在任何 OEM 备份中都不可使用 rate 命令，因此没有方法来调整 RMAN 备份的速度。对于 duration 命令，总是可以指定具有 minimize time 或 minimize load 选项的备份窗口。

RMAN 实习：在 OEM 中配置备份设置

实习注意事项

该实习假定您拥有 v102 数据库，并且要配置该数据库以使用两个信道备份到不同于 FRA 的磁盘位置，需要值为 2 的 filesperset，以及时间为 7 天的恢复窗口。

步骤 1　设置磁盘备份设置。单击希望配置的数据库，然后单击 Availability 选项卡。在 Availability 选项卡上，单击 Backup Settings 链接。显示的默认页面是 Device 选项卡。将 Parallelism 的值改为 2，Disk Backup Location 的值改为/u01/backup。单击右边的 Test Disk Backup 按钮以确认指定的位置存在。

步骤 2　单击 Backup Set 选项卡。将 Maximum Backup Piece(File) Size 改为 500MB。选择 ZLIB 压缩算法。

步骤 3　单击 Policy 选项卡。选中 Automatically Backup the Control File and Server Parameter File 复选框。设置 Autobackup Disk Location 为 u01/backup(与步骤 1 中的备份相同的位置)。

步骤 4　在 Retention Policy 标题的下方，将策略改为第二个选项，即 Retain backups that are necessary for a recovery to any time，并且设置值为 7 天。

步骤 5　在 Host Credentials 下方，设置安装数据软件的用户的主机服务器用户名和口令。然后单击 OK 按钮。

步骤 6　确认改动已经对该数据库生效。连接到驻留该数据库的主机服务器：

```
Export ORACLE_SID=v112
rman target /
RMAN> show all;
using target database control file instead of recovery catalog
RMAN configuration parameters are:
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
CONFIGURE CONTROLFILE AUTOBACKUP ON;
…
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO
'/u01/backup/%F';
…
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO BACKUPSET;
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT    '/u01/backup/%U'
MAXPIECESIZE 500 M;
…
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' MAXPIECESIZE 500 M;
…
```

13.5 配置恢复设置

配置备份设置之后，OEM 提供了页面来配置恢复设置。可以在数据库目标的主 Availability 选项卡中通过单击 Recovery Settings 链接来访问该页面。配置恢复设置实际上包括大量不同的选项。在 Recovery Settings 页面中，OEM 将恢复设置进一步划分为 3 种恢复类型：实例恢复、介质恢复和闪回恢复，下面的小节将介绍这 3 种恢复类型。

13.5.1 实例恢复

只有一个设置涉及实例恢复，即 FAST_START_MTTR_TARGET，如图 13-10 所示。这是以秒为单位确定目标值的初始化参数，当需要在数据库上初始化实例恢复时该值会很有用。MTTR 是 mean time to recover(恢复平均时间)的首字母缩写词，它表示数据库因某些原因崩溃后再次恢复为可操作状态所花费的时间。

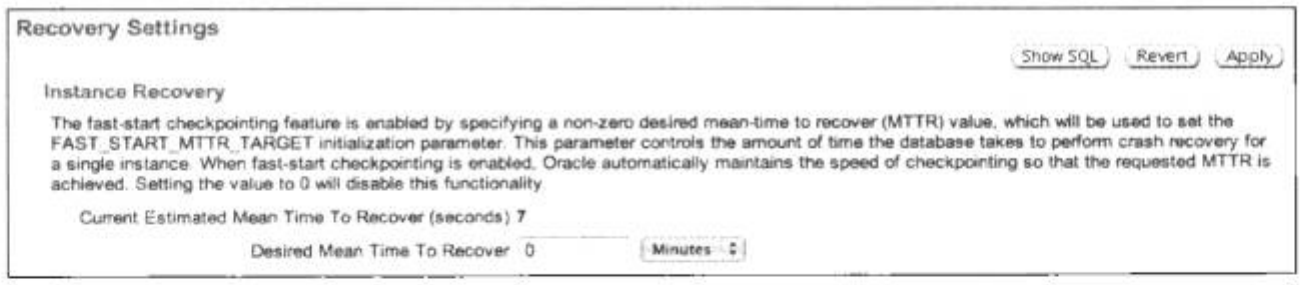


图 13-10 实例恢复设置

数据库在严重崩溃(或异常关闭)后再次启动时，Oracle 会自动初始化实例恢复。实例恢复使用联机重做日志和撤消段来删除在崩溃时没有提交的所有事务，并且为了以后的活动而将数据库设置为经过整理的、同步的状态。实例恢复必须在启动数据库之前完成，从而最小化实例恢复以减少数据库在崩溃后重新变为可操作状态的时间。

FAST_START_MTTR_TARGET 是许多改动的组合，这些改动在内部进行以帮助加速实例的恢复。FAST_START_MTTR_TARGET 也称为自动检查点调整。该参数使得 LOG_CHECKPOINT_TIMEOUT 和 LOG_CHECKPOINT_INTERVAL 参数无效。在 OEM 中，可以以秒为单位指定该值，有效值范围是 0~3600 秒(或者 0~60 分钟)。

13.5.2 介质恢复

在介质恢复的设置页面中有实现 ARCHIVELOG 模式和 NOARCHIVELOG 模式切换的设置。如果没有人告诉您，则很难发现这一点——您可能会开始在数据库目标的 Server 页面中查找 Archive logs 链接。这种模式切换功能位于 Recovery Settings 页面的中间位置，以复选框的形式显示，但是不要认为 OEM 已经激活了 Oracle RDBMS 中的该功能：您仍然必须在该功能生效之前重新启动计算机。实际上，OEM 会提交作业给 OS 以重启数据库并实现该更改。

除了启用 ARCHIVELOG 模式外，也可以在此处指定 LOG_ARCHIVE_FORMAT 和 LOG_ARCHIVE_DEST 参数。在 11*g* 中，参数默认为 USE_DB_RECOVERY_FILE_DEST，也称为闪回恢复区(FRA)。

在 11g 中，添加了一个 Enable Minimal Supplemental Logging 复选框，如图 13-11 所示。它为需要使用 LogMiner(日志分析)实用程序的额外的日志记录配置数据库。

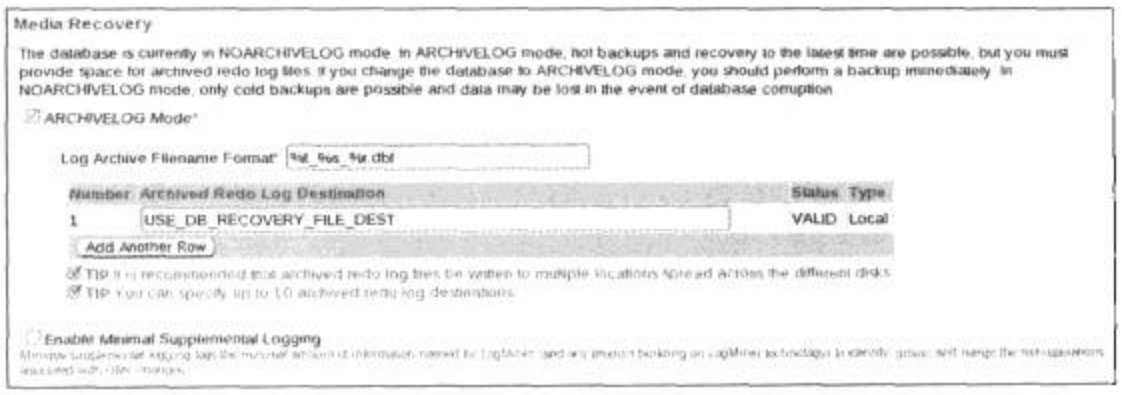


图 13-11　介质恢复设置

13.5.3　闪回恢复

在 Flash Recovery 部分中，我们可以配置 FRA 并且提供 FRA 目标和大小。更为有用(但隐藏在配置页面中)的内容是显示 FRA 中当前已使用空间的饼图(如图 13-12 所示)，该饼图按照文件类型进行划分。这是极好的 FRA 快速视图，在其他位置很难找到该饼图。记住该饼图的位置在闪回恢复配置设置下方，这样可以在后面节省宝贵的时间。

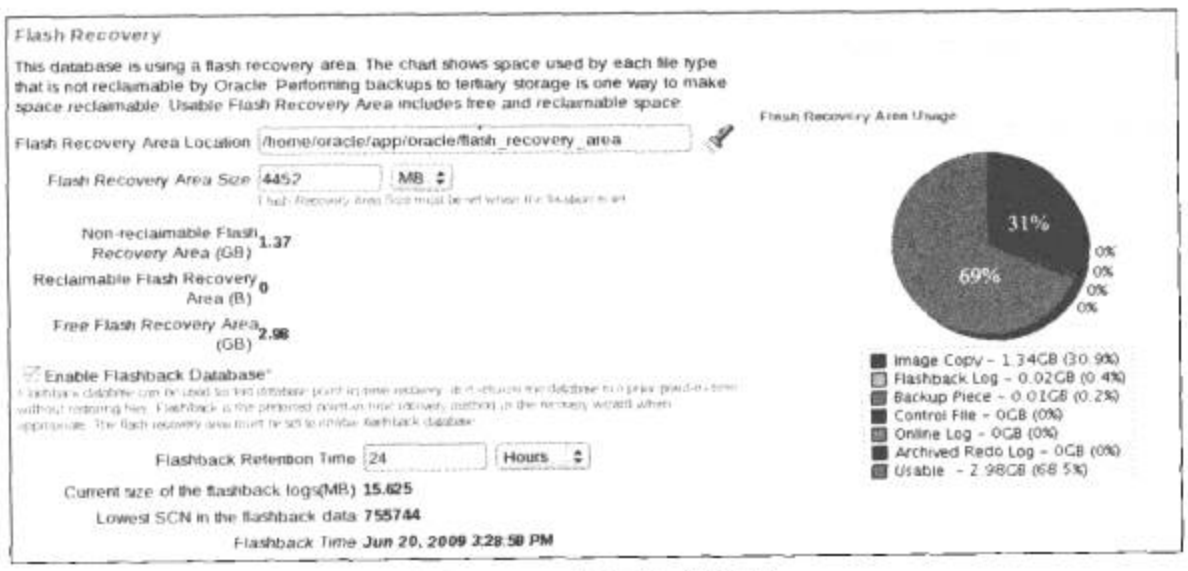


图 13-12　闪回恢复设置

除了配置 FRA 以外，在 Flash Recovery 部分中可以打开 Flashback Database(闪回数据库)，在 10g 中介绍过这个功能，它可以通过回卷数据库(rewinding database)来从根本上减少时间点恢复。第 15 章将讨论闪回数据库和其中需要的所有不同配置需求以及资源。

类似于 ARCHIVELOG 模式，打开 Flashback Database 需要重新启动数据库；我们建议您同时做这两项工作，这样可以省去一些麻烦。打开 Flashback Database 后，可以以小时为单位设置闪回保存时间目标。

最后需要注意的是 Apply Changes to SPFILE Only 复选框。选中该复选框等同于按照如下方式改变参数。

```
alter database set db_recovery_file_dest='/u01/fra' scope=spfile;
```

当然，需要注意选中该复选框将不会“保存”ARCHIVELOG 模式和闪回数据库的改动。可以重新启动数据库并改动它们；或者等以后重新启动数据库后再改动它们。

归档日志和闪回恢复选项有时不可用的原因

如果在 Recovery Settings 页面中发现打开 ARCHIVELOG 模式和修改 LOG_ARCHIVE_DEST 位置的选项在 OEM 中为灰色(禁止改变模式或闪回恢复选项)，则表示不具有 SYSDBA 权限。必须从 Grid Control 或 Database Control 中注销，然后以具有 SYSDBA 权限的 SYS 用户登录，这样才能在 Recovery Settings 页面上进行改动。

RMAN 实习：在 OEM 中配置恢复设置

实习注意事项

该实习实现如下改动：设置除了 FRA 之外的第二个归档目标，向 FRA 增加更多的空间，并且启用闪回数据库。该实习假定已经处于 ARCHIVELOG 模式(如果没有处于该模式，请参见第 3 章，或者在 Recovery Settings 页面中选择 ARCHIVE Log 复选框)。

步骤 1 作为具有 SYSDBA 角色的 SYS 用户登录到 OEM。如果没有作为 SYS 登录，Recovery Settings 页面上的选项将不可用。

步骤 2 定位到 Database|Availability|Recovery Settings 页面。

步骤 3 设置新的归档日志目标为/u01/backup。单击 Add Another Row，然后指定位置(这里我们使用/home/oracle/backup)。

步骤 4 将闪回恢复区的大小改为 4GB。

步骤 5 选中 Enable Flashback Database 复选框。设置闪回保存时间为 24 小时。

步骤 6 单击 Apply 按钮。这时会提示重新启动数据库。单击 Yes 按钮。然后需要为关机提供主机凭证和数据库凭证。

步骤 7 数据库重启后，回到 Recovery Settings 页面确认发生的更改。

13.6 在 OEM 中配置恢复目录

企业管理器实际上不能创建恢复目录。您仍然需要手动创建目录用户，授予该用户 RECOVERY_CATALOG_OWNER 角色，然后连接 RMAN 与该用户，并且执行 create catalog 命令。对于 OEM 中的这些步骤，没有任何向导或相关内容。在 Recovery Catalog Settings 页面有一个 Add Recovery Catalog 按钮。然而，这个按钮只用来添加已存在的恢复目录到企业管理器存储库中；即让 EM 知道恢复目录的存在，以便它可以添加特定的数据库到指定的目录中。

一旦创建了目录，就可以通知 OEM：您希望使用恢复目录。在 OEM 中注册了恢复目录后，就可以在该目录中注册目标。如果已经注册了多个恢复目录，就可以指定在不同的备份和恢复操作期间使用特定的恢复目录。

RMAN 实习：在 OEM 中注册恢复目录

实习注意事项

该实习在数据库 emrep 中手动创建目录，然后在 OEM 中注册它以进行使用。

步骤 1　作为用户 SYS 连接到存储库数据库，并且创建用于目录的用户：

```
SQL> create tablespace reco_cat datafile
'/u01/app/oracle/oradata/emrep/reco_cat1.dbf' size 100m;
Tablespace created.
SQL> create user rman identified by rman
  2   default tablespace reco_cat
  3   quota unlimited on reco_cat ;
User created.
SQL> grant connect, resource, recovery_catalog_owner to rman;
Grant succeeded.
SQL> connect rman/rman
Connected.
```

步骤 2　在执行这个步骤之前确保路径为 11.2 $ORACLE_HOME/bin，从而 RMAN 可执行程序是 11.2 版本。否则，OEM 将不能够注册目录!

```
$ echo $PATH
/u01/app/oracle/product/10.2.0/bin
  [oracle@dex oracle]$ rman catalog rman/rman@v112
Recovery Manager: Release 11.2.0.0.2 on Wed Dec 21
Copyright (c) 1982, 2009, Oracle. All rights reserved.
connected to recovery catalog database
RMAN> create catalog;
recovery catalog created
```

步骤 3　现在已经创建了目录，接下来进入到 OEM | Database | Availability | Recovery Catalog Settings 页面。选择 Use Recovery Catalog，然后单击 Add Recovery Catalog 按钮。在 Grid Control(如图 13-13 所示)中，可以选择已经发现的数据库或提供主机、端口、SID 以及 RMAN 用户名和口令。在 Database Control 中，不可以从列表中选择，并且必须提供主机:端口:sid 的组合。而且，需要提供恢复目录凭证(RMAN 用户在第二步中创建的)。

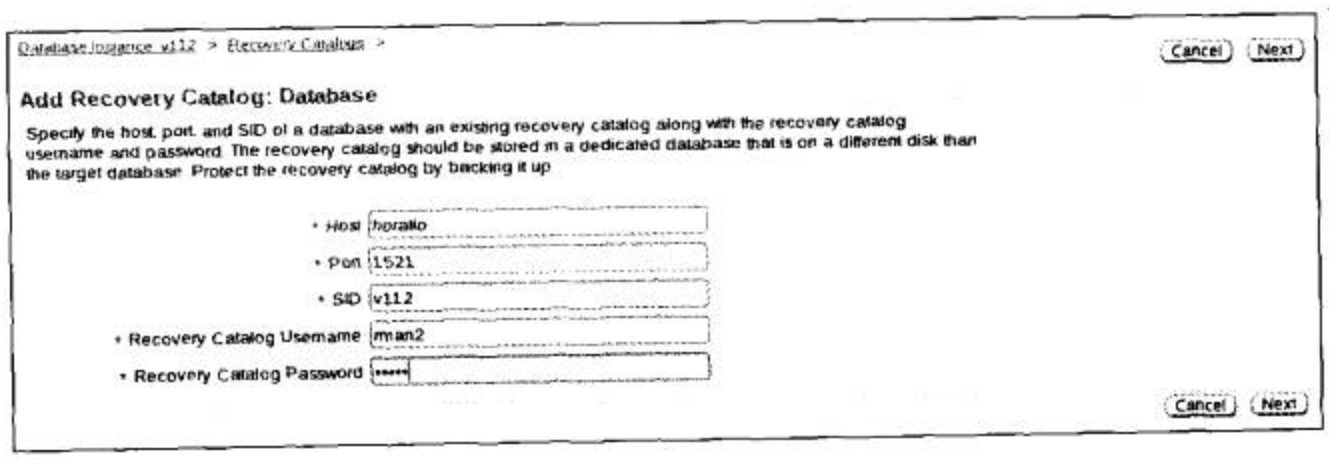


图 13-13　Add Recovery Catalog: Database 页面

步骤 4　在对 Add Recovery Catalog 页面进行操作之后，转到了 Recovery Catalog Settings 页面，该页面要求再次输入 Recovery Catalog 的用户名和口令，以及服务器的主机凭证，如图 13-14 所示。这些步骤完成之后，单击 OK 按钮，就完成了注册数据库的操作。

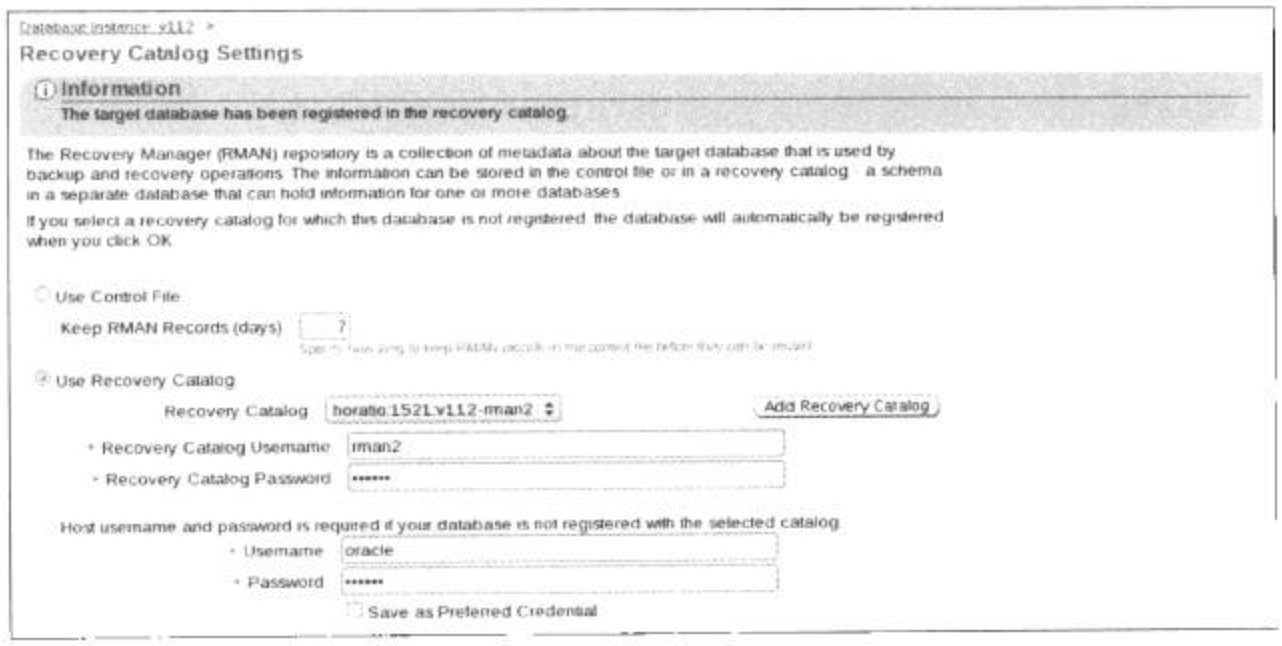


图 13-14　Add Recovery Catalog Settings 页面

Recovery Catalog Settings 中的 Related Links

在 Recovery Catalog Settings 页面的下部有三个“相关链接(related links)”，它们提供了额外的功能：

- Resynchronize Catalog：手动完成同步操作，使数据库控制文件和恢复目录进入同步发生状态。
- Recovery Catalogs：浏览和管理所有 EM 已知的恢复目录。
- Unregister Database：从选中的恢复目录中永久的删除某数据库。

13.7　企业管理器中的数据库备份

现在已经为 OEM 界面中的备份配置了数据库，接下来可以实际地进行备份。选择数据库并单击 Availability 选项卡后，在 OEM 控制台中单击 Schedule Backup 链接。在 Schedule Backup 页面上，提供了两个选项：Oracle-Suggested Backup 和 Schedule a Customized Backup 选项。

13.7.1　Oracle 建议备份策略

从 Oracle Database 10g 开始，Oracle 集成了可以直接使用的完整备份策略。但只可以通过 OEM(在 Grid Control 和 Database Control)使用这种备份策略，并且需要具有 FRA。这种由 Oracle 建议的备份策略将检查已经为数据库配置的备份和还原设置，并给出是使用磁盘备份、磁带备份还是结合两者备份的结论。

1. 只有磁盘的 Oracle 建议备份策略

只有磁盘的备份策略是最为直观的备份策略，它使用“增量应用于数据库副本”功能来创建自动整理的备份策略。下面是使用该备份策略的示例。在该示例中，我们在星期一晚上的 10 点启动 Oracle 建议备份策略。数据库运行在 ARCHIVELOG 模式中，启用了自动控制文件备份和自动数据库参数文件备份，并且使用了 FRA。

- **星期一晚上**：针对数据库获得每个数据文件的完整映像副本。该备份存储在 FRA 中。
- **星期二晚上**：创建 1 级的增量备份集。备份从星期一晚上 10 点以来已经更改的所有数据块。
- **星期三晚上**：首先，RMAN 将星期二晚上的增量备份应用于星期一晚上的映像副本，这就使完全镜像副本备份成为星期二晚上 10 点时的数据库的当前副本。完成这项工作后，RMAN 获得新的 1 级备份。
- **星期四晚上**：RMAN 将星期三晚上的增量备份应用于镜像副本，这就使数据库备份成为星期三晚上 10 点时的数据库的当前副本。然后，获得新的 1 级增量备份，该增量备份保存了从星期三晚上以来的所有更改。
- **星期五晚上**：RMAN 执行与星期四晚上相同的工作，在周六也执行相同的工作，星期天也是如此。每天重复这些工作，直到您告诉 RMAN 停止该操作。

关于这种备份策略有一些需要注意的内容。首先，只获得一次完全数据库备份。在此之后，RMAN 只获得 1 级增量备份。然后，在每天晚上应用前一天晚上的增量备份。其次，该策略确保数据库备份至少是当前时间点之后 24 小时的备份。这就允许通过时间点恢复来恢复到之前 24 小时内的任何时间点，但是不能恢复在此之前的任何内容。数据库备份最多是当前时间之后 48 小时的备份，因此不能恢复超过这个时间的内容。

该策略有一些限制，这也是“一体适用”式备份方法的一种缺陷。

2. 只有磁带的 Oracle 建议备份策略

只有磁带的建议策略在许多方面不同于只有磁盘的建议策略。首先，在 FRA 中不会创建任何备份。当然，在 FRA 中会累加归档日志，但是所有备份将直接存入磁带设备。此外，只有磁带的策略不能利用应用增量备份的特性。记住，对磁带不可以使用映像副本备份。所有磁带备份都是备份集类型的备份。

当调度只有磁带的 Oracle 建议备份时，将生成两个 RMAN 脚本：每日的备份和每星期的备份。首先，运行每星期的脚本，这会创建一个完全数据库备份，包括所有归档日志。然后，运行每日的脚本。每日的备份只对更改的数据块和没有备份的所有归档日志进行增量备份。然后，经过一个星期后，再次运行完全数据库备份。

在使用只有磁带的备份向导期间，如果没有指定保留策略，OEM 会要求您指定一个保留策略。然后，作为每日备份的一部分，在磁带信道上实施保留策略。

生成只有磁带的建议策略后，将会发现脚本可能类似于如下。

```
Daily Script:
```

```
run {
allocate channel oem_sbt_backup1 type 'SBT_TAPE' format '%U' parms
'nb_ora_server=(horatio.hadba.com)';
allocate channel oem_sbt_backup2 type 'SBT_TAPE' format '%U' parms
'nb_ora_server=(horatio.hadba.com)';
backup incremental level 1 cumulative database;
backup archivelog all not backed up;
}
allocate channel for maintenance device type 'SBT_TAPE' parms
'nb_ora_server=(horatio.hadba.com)';
delete noprompt obsolete recovery window of 7 days device type 'SBT_TAPE';

Weekly Script:
run {allocate channel oem_sbt_backup1 type 'SBT_TAPE' format '%U' parms
'nb_ora_server=(horatio.hadba.com)';
allocate channel oem_sbt_backup2 type 'SBT_TAPE' format '%U' parms
'nb_ora_server=(horatio.hadba.com)';
backup incremental level 0 database;
backup archivelog all not backed up;
```

3. 结合磁盘和磁带的 Oracle 建议备份策略

当结合磁盘备份与磁带备份时，这种混合的解决方案比前面任何一种 Oracle 建议策略都需要用户输入更多的内容。并且，在第一次使用该备份策略时，就必须决定输入的配置的内容。

首先，考虑该策略的磁盘部分。结合磁盘和磁带的策略中的磁盘备份与只有磁盘的策略相同：在 FRA 中存储完全镜像副本，然后将每晚的增量备份应用于映像副本。

接下来考虑策略的磁带部分。必须决定每天向磁带上备份多少内容。对于按星期执行的备份，则建议的策略是备份所有与恢复相关的文件(在 RMAN 中使用 backup recovery files 命令)。但是，必须选择每天备份多少内容：不备份任何内容、只备份归档日志、备份归档日志和增量备份、或者备份归档日志和完全数据库副本。

如果决定每天将归档日志和增量备份到磁带，下面是 OEM 构建的每日脚本和每星期脚本：

```
Daily Script:
run {
allocate channel oem_disk_backup device type disk;
recover copy of database with tag 'ORA$OEM_LEVEL_0';
backup incremental level 1 cumulative
copies=1 for recover of copy with tag
'ORA$OEM_LEVEL_0' database;
release channel oem_disk_backup;
allocate channel oem_sbt_backup1 type 'SBT_TAPE' format '%U' parms
  'nb_ora_server=(horatio.hadba.com)';
backup archivelog all not backed up;
backup backupset all not backed up since time 'SYSDATE-1';
}
```

```
allocate channel for maintenance device type 'SBT_TAPE' parms
  'nb_ora_server=(horatio.hadba.com)';
delete noprompt obsolete recovery window of 7 days device type
'SBT_TAPE';

Weekly Script:
run {
allocate channel oem_disk_backup device type disk;
recover copy of database with tag 'ORA$OEM_LEVEL_0';
backup incremental level 1 cumulative copies=1 for recover of copy with tag
  'ORA$OEM_LEVEL_0' database;
release channel oem_disk_backup;
allocate channel oem_sbt_backup1 type 'SBT_TAPE' format '%U' parms
'nb_ora_server=(horatio.hadba.com)';
backup recovery area;
}
allocate channel for maintenance device type 'SBT_TAPE' parms
'nb_ora_server=(horatio.hadba.com)';
delete noprompt obsolete recovery window of 7 days device type 'SBT_TAPE';
```

4. Oracle 建议策略的注意事项

使用已有的默认备份策略既有优点也有缺点。其优点在于能够“设置和忘记”备份——这些备份操作将永久运行并且自动清理。使用默认的备份策略可以提供一点保证，特别是对于一些由一般人操作的低优先级开发数据库。如果他们的操作给您带来很多麻烦，您总是可使用这些备份来修正问题并且使每个人免于大量复杂的工作。

但是，默认的备份策略几乎不包含任何自定义设置，并且有时这种默认的备份策略与您的 SLA 不符。这点从它执行时间点恢复来恢复到过去某些时间点的例子上(只有基于磁盘的策略有此限制)可以看出来。记住，这些策略仅仅是 Oracle 建议的策略，而不是必须使用的策略。如果这些策略给您带来许多麻烦，简单的解决方法是删除建议的策略并构建自己的备份策略。

13.7.2　调度自定义的备份

在 Availability 页面中除了可以选择调度 Oracle 建议的备份策略，还可以选择构建自己的自定义备份作业。这是一个向导驱动的过程，用户可以逐步地对备份设置进行各种选择。当结束该向导时，可以作为一次性的备份立刻运行该备份，在后面对其进行调度，或者对其进行设置以经常重复运行该备份。

首先，花费一些时间来提前开发完整备份策略非常重要，但是 OEM 将提供关于如何开发这种策略的一些指导。这些提示以小字体的形式显示在一些选项的下方，在备份调度向导中需要在这些选项中进行选择。

备份调度向导分为如下 4 步：

(1) 选择备份选项(即备份哪些内容)。

(2) 选择备份设置(即如何备份)。

(3) 调度备份(即何时开始备份)。

(4) 执行最终的检查(是否对所有选择感到满意)。

在最终检查页面中，将会显示向导创建的 RMAN 脚本，如下面的示例所示。在这个阶段中，可以提交要执行的作业，或者在此处编辑 RMAN 脚本，从而对脚本进行较少的调整以达到满意的程度。注意，如果决定手动修改脚本，OEM 会警告您：您将不能够返回并在备份向导页面中做任何更改。亲自更改完之后，就可以取消这些更改或提交作业。

13.7.3 RMAN 脚本作业与调度备份向导

可以使用另一种方法来使用 OEM 调度和运行目标数据库的 RMAN 备份。如果进入到 Grid Control 的 Jobs 选项卡(或者 Database Control 中 Related Links 下的 Jobs 链接)并且查看可创建作业类型的下拉列表框，就会看到 RMAN 脚本。这是特定类型的作业规范，允许您使用 OEM 执行目标服务器上已有的 RMAN 脚本。

因此，RMAN 脚本作业和主机命令作业之间的区别在于：两者都要求在调度作业之前，脚本已存在。但是如果选择 RMAN 脚本作业，OEM 就会使用它的内置机制来确保在运行该脚本之前目标数据库环境已正确配置。这是非常优秀的特性，如果您曾经遇到过关于 RMAN 可执行程序与目标数据库的“兼容性矩阵”问题，就会了解该特性的作用。

OEM 总是尝试分配磁带信道给维护操作的原因

您可能已经注意到，在使用 OEM 向导调度备份之后检查 RMAN 脚本时，OEM 具有如下代码行:

```
allocate channel for maintenance type 'SBT_TAPE';
delete noprompt obsolete device type disk;
```

这段代码的作用是什么？如果在任何时候进入到 Configure Backup Settings 页面并且设置磁带驱动器的数量为某些值，则 OEM 就会进入目标数据库并在永久性配置参数中配置磁带信道。因此，它假设任何维护操作都应该总是检查磁带和磁盘。在 OEM 中，没有方法改变这一点。您必须进入 RMAN 命令行并输入如下代码:

```
configure channel device type 'sbt_tape' clear;
```

RMAN 实习：在 OEM 中创建 RMAN 脚本作业

实习注意事项

该实习使用 OEM 调度 RMAN 备份。然而，该备份将由一个 RMAN 脚本指定，这个 RMAN 脚本文件位于具有数据库 v112 的 Linux 计算机上。在这个示例中，该脚本位于/home/oracle/scripts 中，名称为 rmanback.rcv，并且包含如下代码:

```
backup incremental level 1 cumulative device type disk
  filesperset = 2 database;
backup device type disk filesperset = 2 archivelog all not backed up
  delete all input;
delete noprompt obsolete device type disk;
```

步骤 1　进入到 OEM 的 Jobs 页面，在 Grid Control 中，右上方有一个 Jobs 选项卡。在 Database Control 中，在 Related Links 底部查找 Jobs 链接。

步骤 2　使用下拉列表来创建类型为 RMAN Script 的新作业(默认为 OS 命令)，指定 RMAN 脚本，然后单击 Go 按钮。这将进入到一个具有多个选项卡的页面。

步骤 3　在 General 选项卡上，将作业命名为 RMAN_BACK_INC_1_ARCH_DELETE_OBS，如图 13-15 所示。在这个选项卡上，还需要添加需要备份的目标(针对 Grid Control)。单击 Add 按钮并选择所需的数据库。

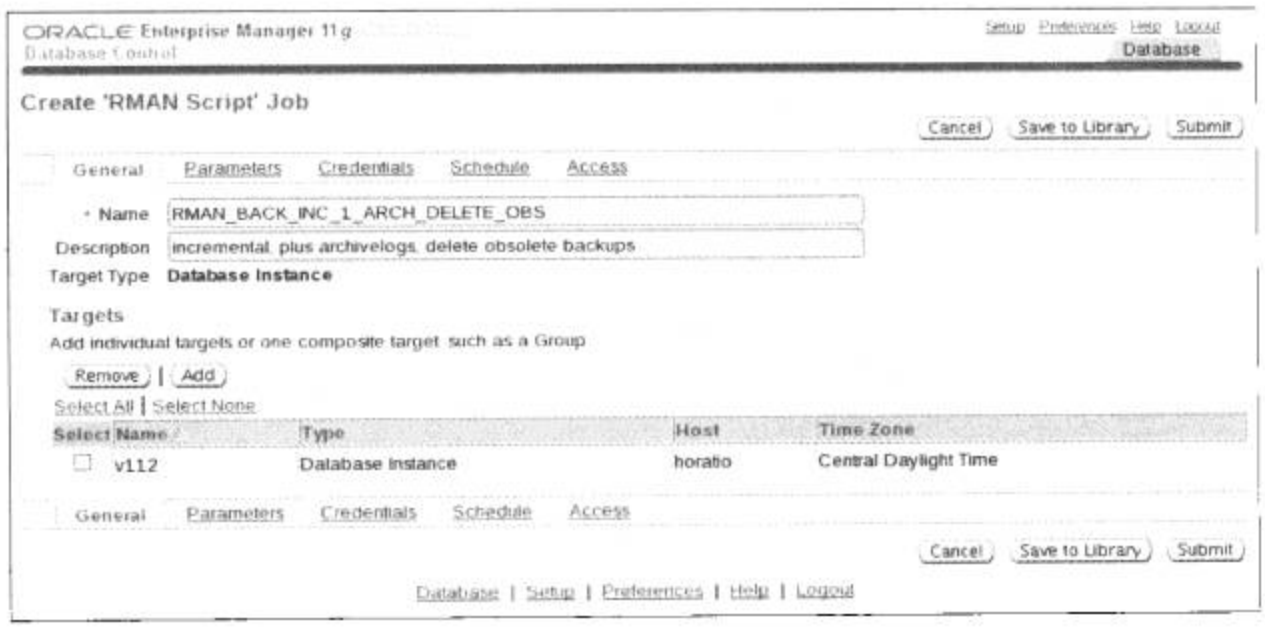


图 13-15　General 选项卡

步骤 4　单击 Parameters 选项卡(如图 13-16 所示)。在其中的文本框中，提供在目标服务器上创建的备份脚本的完整路径和名称。

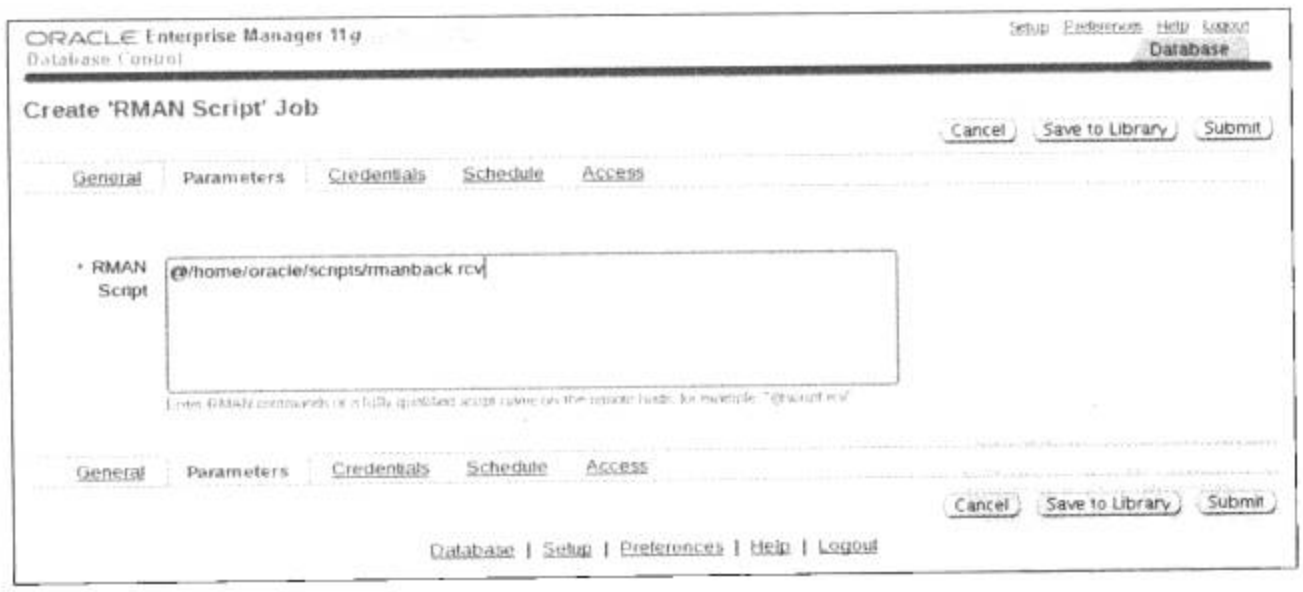


图 13-16　Parameters 选项卡

步骤 5　在 Credentials 选项卡上(如图 13-17 所示)，确保选择了 SYSDBA 数据库凭证，这是因为 RMAN 需要该凭证。

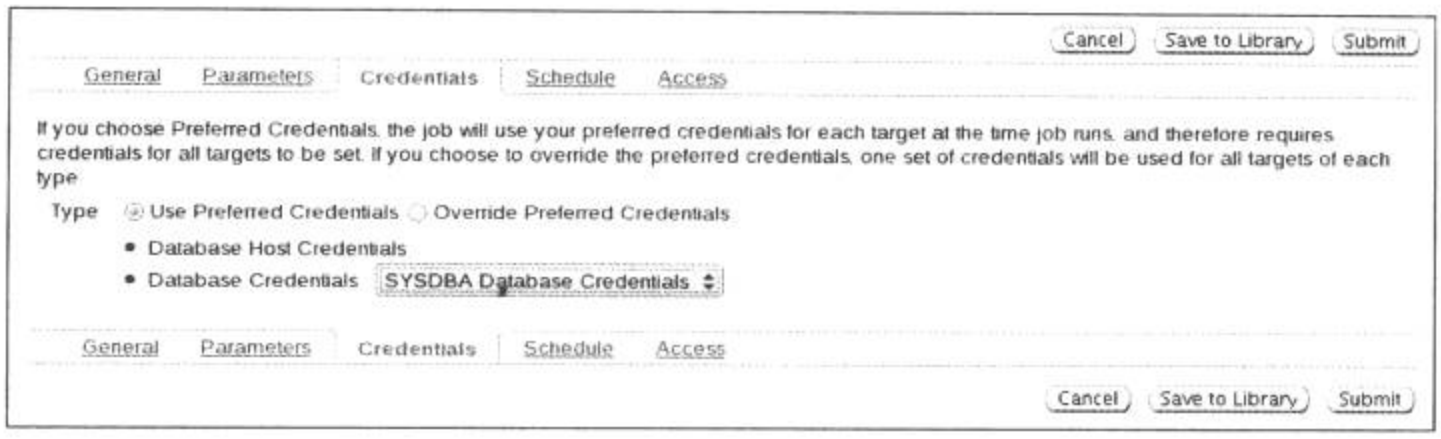


图 13-17 Credentials 选项卡

步骤 6 在 Schedule 选项卡上(如图 13-18 所示)，为作业设置时间表。可能的选择是 One Time(Immediately)、One Time(Later)或 Repeating。在新的 11g 中，也可以为备份作业指定 Grace Period: 通过未定义指定，告诉 EM 让作业一直运行，直到出现作业结束信号为止。通过选择 End After，指示 EM，如果作业在指定的时间没有完成，则让作业中断。

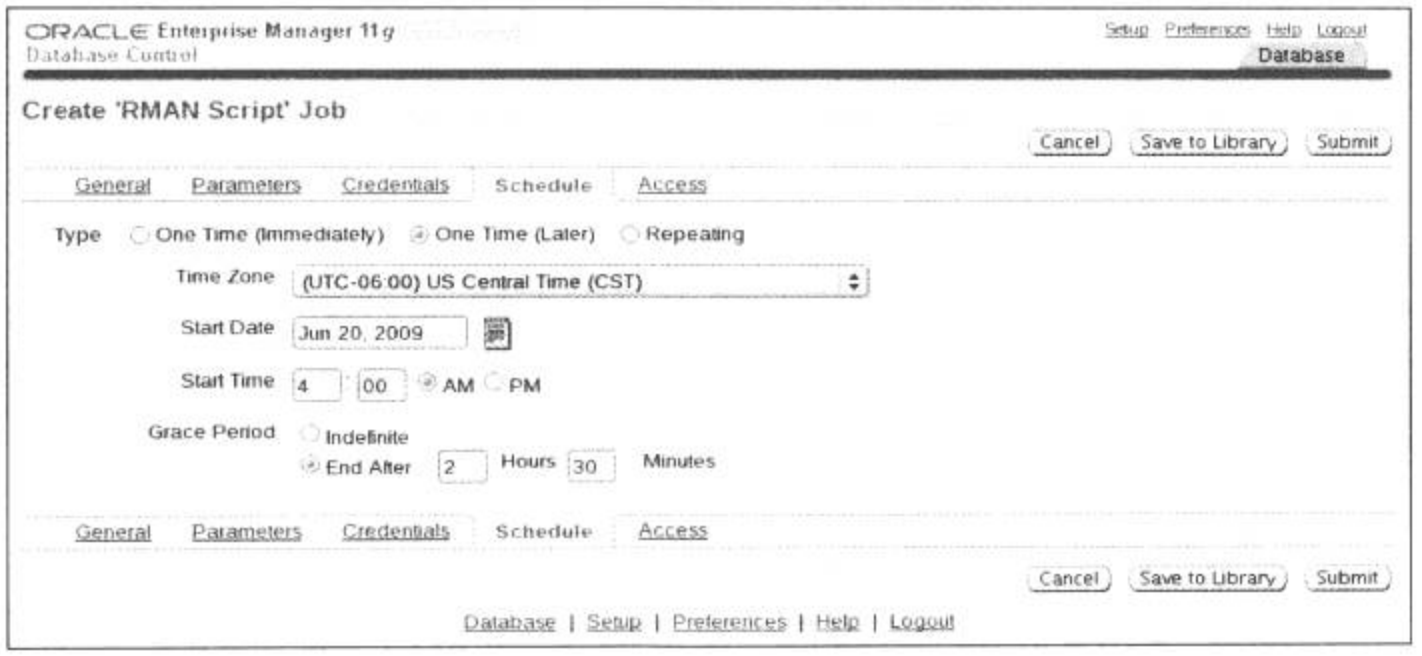


图 13-18 Schedule 选项卡

步骤 7 单击页面顶部的 Submit 按钮，发送作业给活动的作业页面。这样作为一次性事件提交作业。也可以不立即提交而将作业保存到库中，或者现在提交作业并将其保存到库中。

13.8 在企业管理器中执行恢复操作

在 OEM 中完成所有备份后，也可以在其中执行恢复操作。这也就是说，很少会看到使用 OEM 的恢复操作——即使是使用 OEM 进行备份。可根据是否能够减轻在遇到数据库停机时产生的恐慌来决定是否使用 OEM 进行恢复。

OEM 不会帮助控制这种恐慌。您可能认为 GUI 是避免产生恐慌的最佳防御工具——易于使用、快速实现、并且非常有效。但是，我们见到的所有 DBA 在恢复情况中都不会

这样认为。在我们参与的最为复杂的恢复情况中，DBA 在遇到困难无法解决时所需的不是易于使用、内置的智能等功能。每个 DBA 在此时所需要的是控制，他们需要对可能控制的每件事情的完全控制。数据库恢复并不是单击大的红色按钮并且期待获得最好的结果。

对于在开发中部署的所有数据库，OEM 中的恢复向导对于负责管理这些数据库的人一般非常有用。换句话说，OEM 恢复可用于拥有 Oracle 建议备份策略的数据库。

但也有例外，例如若已经了解了 OEM 恢复的工作原理，则会注意到它可以将您导向 Oracle 闪回技术，第 15 章将对闪回技术进行讨论。闪回技术提供了针对停机的全新工具集，特别是针对用户错误导致的数据损坏。因此，如果 OEM 恢复是您正在寻找的控制方法，则必须声明的一点是：如果未研究闪回技术的选项，则您对恢复情况的控制还不完全。

然而，恢复向导并不能在出现未知故障之后帮助您解除恐慌，在 11gR2 中有个新的功能，它将帮助您预先地知道故障，因此完全让您免于恐慌。这个新实用程序成为了 11.2 数据库的一部分，它被称为 Data Recovery Advisor(数据恢复顾问)。

13.8.1　数据恢复顾问和 OEM 检查器

数据恢复顾问(DRA)是数据库的核心 RDBMS 中构建的顾问功能。它利用与已经在 11g 中建立的用来合并所有诊断和故障信息相同的诊断存储库(Diagnostic Repository)，以便进行更多的程序性的修复(更多自动诊断存储库的内容，请参见第 3 章)。DRA 在诊断存储库中挖掘数据，然后决定 DBA 是否应该为了修复数据库并保持数据库的可用性而执行一个恢复操作。

为了在恢复情况中提供智能操作，DRA 必须可以访问诊断存储库中的额外数据。如果 RMAN 在最后的备份操作中发现任何损坏块，一些信息例如数据文件的已知状态，无论如何将在存储库中清除。然而，只要运行新集成的完整的数据库健康检查器，有些信息将保存。这些检查器在数据库的不同部分执行诊断数据采集合操作，然后把不同操作者(例如 DRA)所执行的操作的数据存储到诊断存储库中。

在 11.2 版本中，健康检查器和 DRA 是 RDBMS 固有的组件，所以不需要使用 EM 来实现这些功能。关于命令行 DRA 在第 3 章有详细的讨论。这里将集中于介绍 EM 界面的工具。GUI 界面有它值得使用的一部分——调度健康检查器操作并给出修复任何中断操作的一站式位置。

1. 在 EM 中的检查器

在 11.2 版本中有七个检查器。每个检查器都对不同的数据库对象执行操作，或者带有不同的体系结构需求。接下来会对他们中的每一个进行描述。他们的共同点是都可以通过向下滚动到 Related Links 链接并单击 Advisor Central 按钮，在主数据库主页上的 EM 中找到它们。Advisor Central 提供两个页面——Advisors 和 Checkers 页面。Advisors 页面是默认视图；需要单击 Checkers 选项卡才能看到健康检查实用程序。

如图 13-19 所示，在 Checkers 页面上，可以看到检查器最近的运行及结果。单击 Run Name 选项，可以到达 Findings 页面，查看是否产生任何要求执行操作的警告日志。每个检查器需要运行唯一的输入，但是它们都共同要求至少两项：必须要为启动的运行命名(名字必须唯一)，如果检查器没有完成操作，必须在 EM 中止检查器后提供暂停。

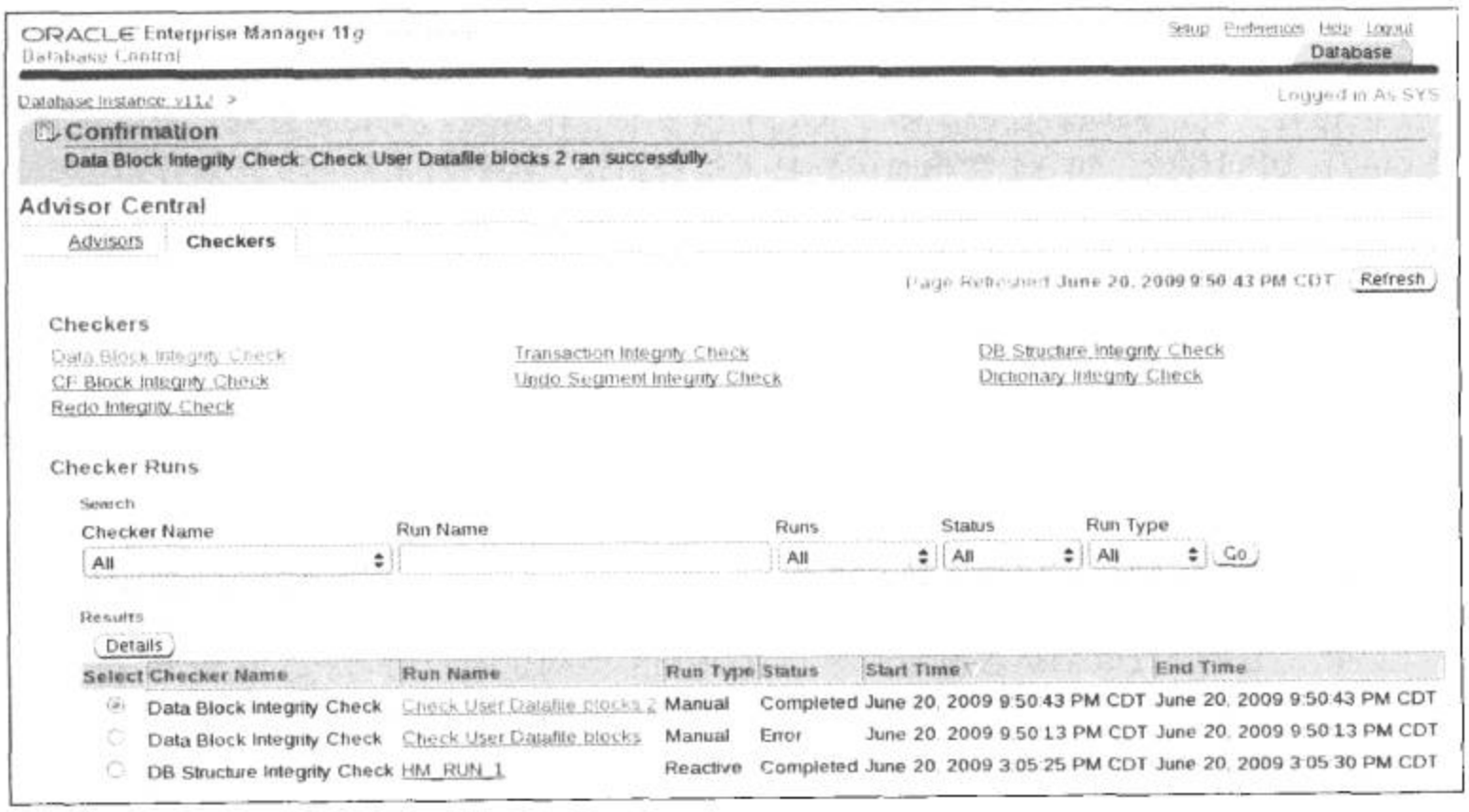


图 13-19 Checkers 页面

- **Data Block Integrity 检查**：检查损坏的单个数据块。需要知道可疑文件和块的数目(例如 ORA-1578 的错误报告)来运行检查器。
- **CF Block Integrity 检查**：它是控制文件检查器。此外，该工具的期望是知道需要完整检查的可疑回块数量。
- **Redo Integrity 检查**：联机检查或者检查归档重做损坏及可达性。指定想从哪个 SCN 开始执行操作，EM 将从那里开始执行检查。
- **Transaction Integrity 检查**：它与 Undo Segment Integrity 检查一样，检查逻辑撤消损坏，除此之外，它进行简单事务处理。需要输入的是事务的 ID。
- **Undo Segment Integrity 检查**：为整体 Undo 部分检查逻辑撤消损坏，需要输入的是 Undo 部分的数量。
- **DB Structure Integrity 检查**：该检查器为数据库文件的可达性，文件头损坏和文件及控制文件之间的 SCN 同步性检查提供提前预警机制。在这里不需要输入任何内容。
- **Dictionary Integrity 检查**：此检查器检查核心数据字典项来确保一致性和完整性。需要提供要检查的数据字典的表的名字和检查器的掩码。

2. 数据恢复顾问

和健康检查的实用程序一样，可以通过数据库主页上 Related Links 中的 Advisor Central 链接到数据恢复顾问。如果在数据库中没有最近相关的恢复结果，则 DRA 将显示一个搜索栏，来搜索以前编写目录的故障。如果一个新的故障存在而且没有修复，则将选择选项 Select failures and...来执行操作，如图 13-20 所示。

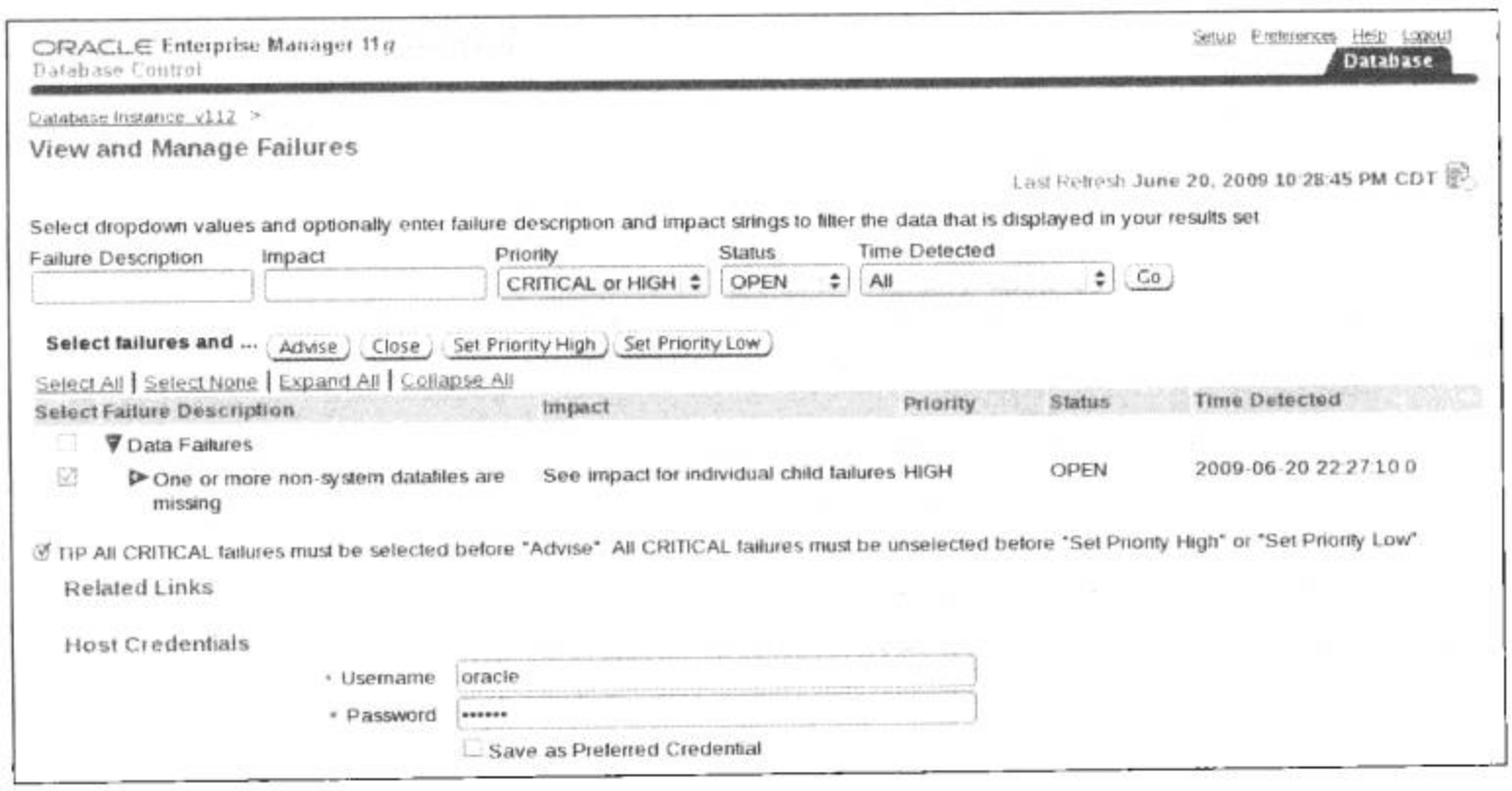


图 13-20　View and Manage Failures 页面

如果单击 Advise 按钮，EM 将用几分钟来验证信息，然后提出怎样处理出现的特殊故障的建议。如果有可自动执行的操作，将征得同意后执行操作。除此之外，将要求执行手动操作，如图 13-21 所示。

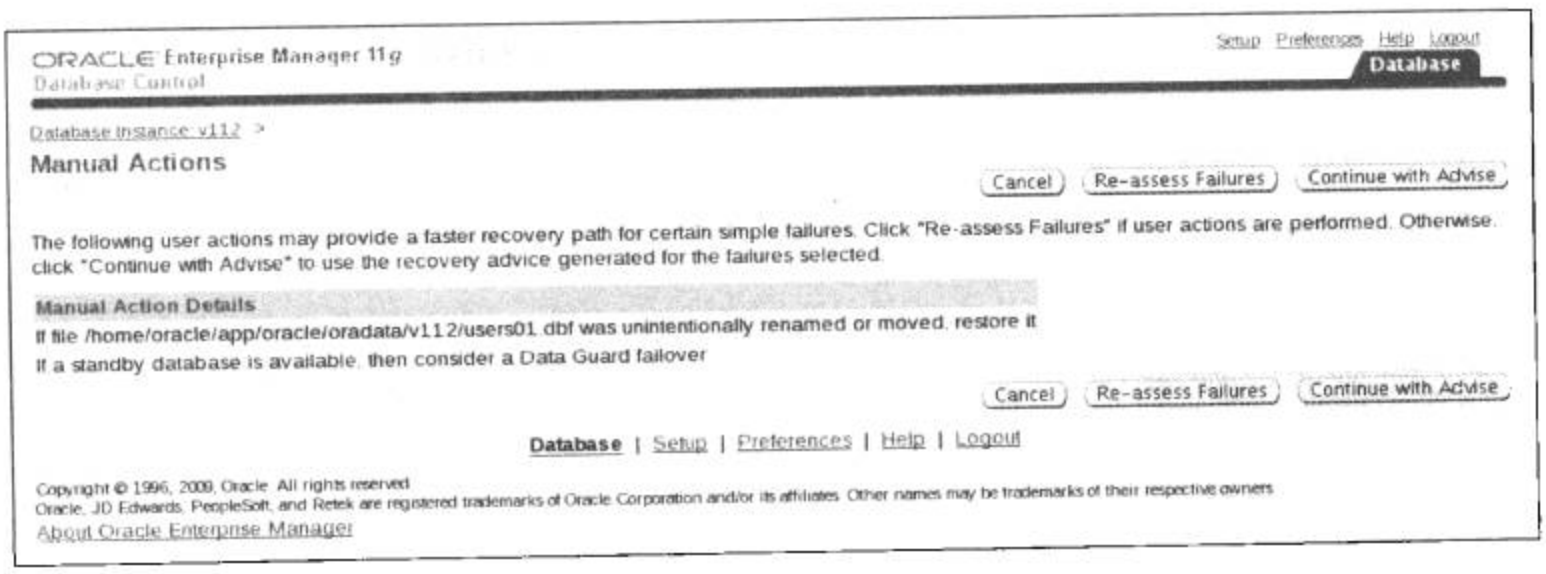


图 13-21　Manual Actions 页面

建议的恢复操作在技术上是完全正确的，但是不能确保它是最好的路径。如之前所述，不能 100%控制，因为 Oracle 不可能知道所有的细节。例如，如图 13-22 所示，文件已

经被重命名了。在为一个文件执行还原操作之前，理应要在服务器上手动执行一些检查并查看是否可以用一种更直接的方式修复问题。除此之外，将根据建议执行操作，还原文件，然后恢复文件，如图 13-22 所示。

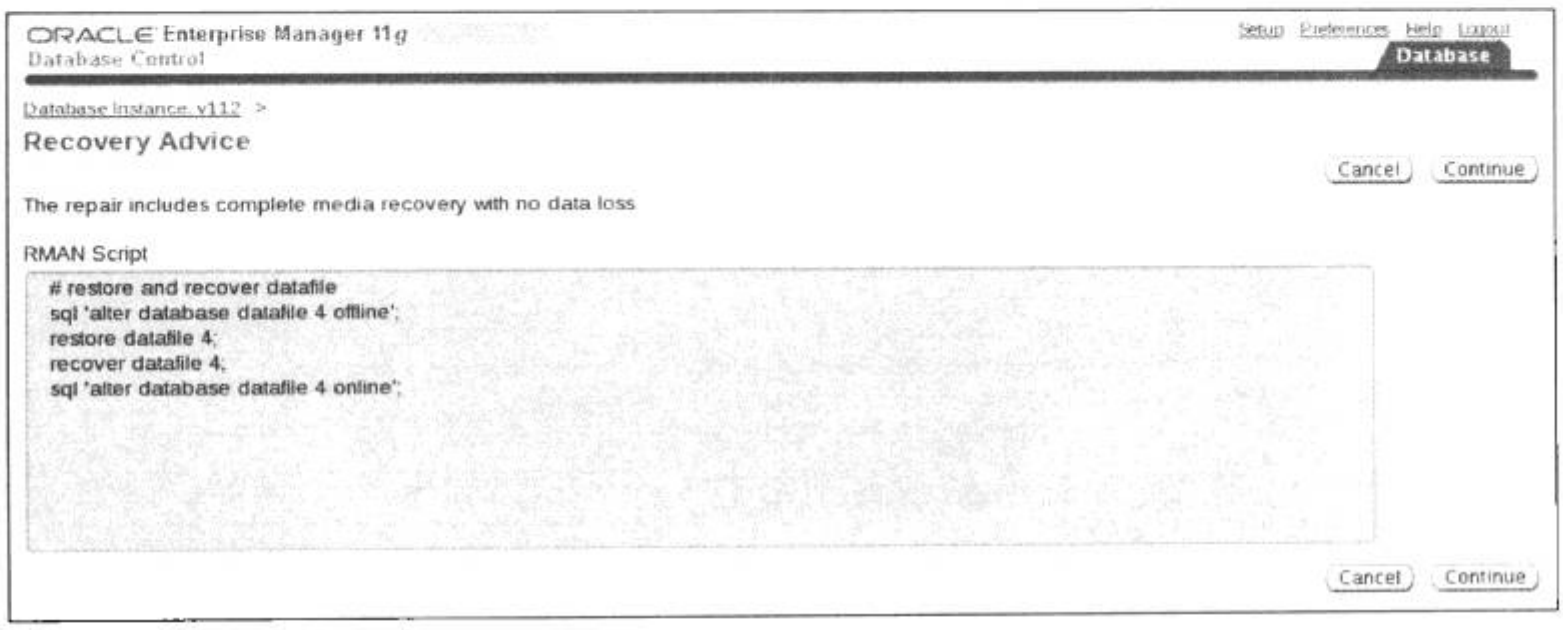


图 13-22 Recovery Advice 页面

最后，在显示了可能性操作之后，数据恢复顾问将按照它提出的建议准备提交恢复作业。如图 13-23 所示，可以看到可能产生的影响(至少程序上是可行和安全的)，可以看到即将运行的 RMAN 脚本，然后单击 Submit Recovery Job 按钮让操作继续运行。在此之后，像在 EM 中跟踪任何其他的作业一样跟踪这个作业，然后到达 Jobs 页面，可以看到该作业的进程，如图 13-23 所示。

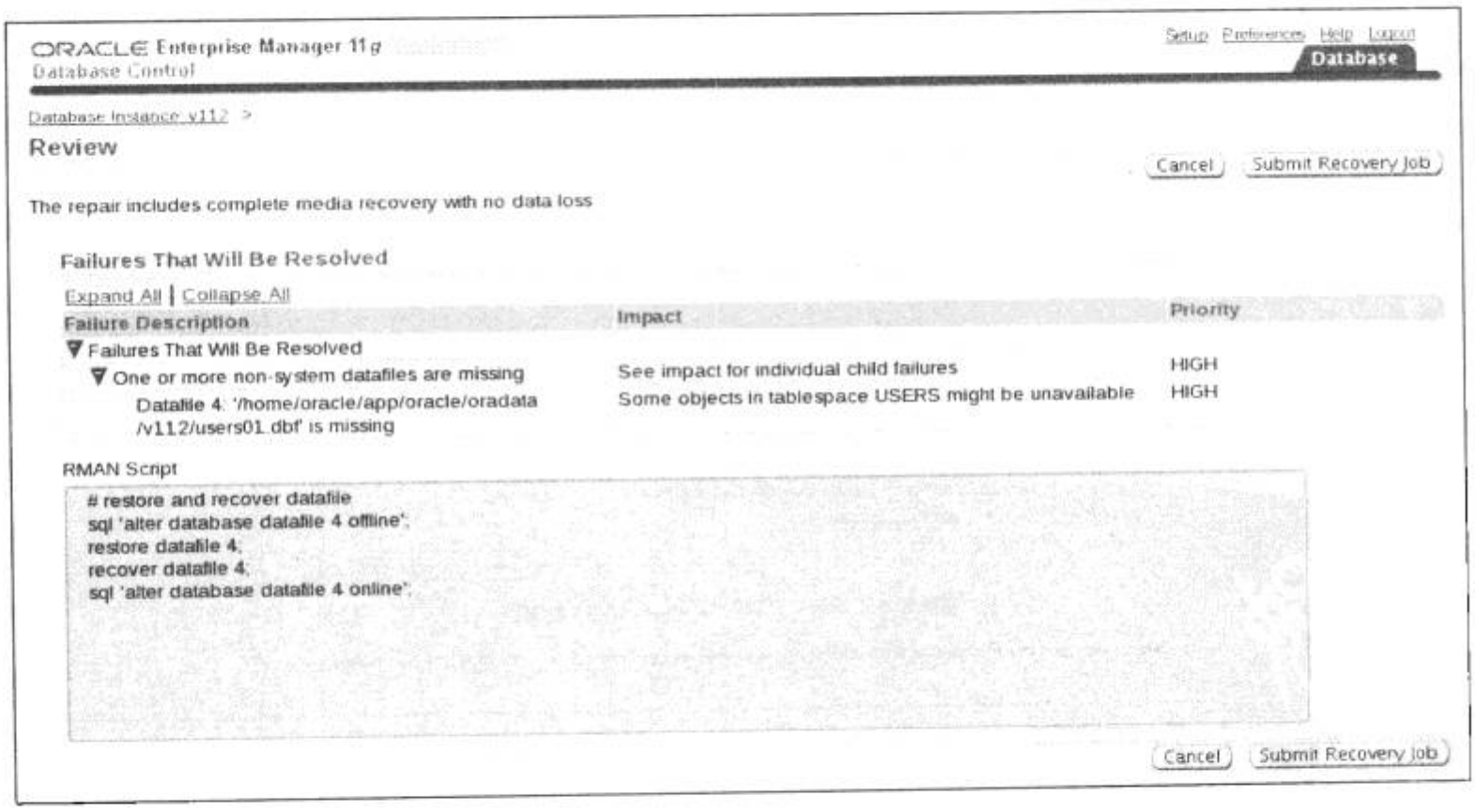


图 13-23 Review 页面

3. 执行恢复页面和数据恢复顾问

当单击 Perform Recovery 页面的，如果数据恢复顾问检测到了故障，将出现标题为“Oracle Advised Recovery”的部分。该部分主要描述检测到的故障并提供 Advise 和 Recover 按钮。选择按钮转到 ADR 页面并通过向导进行指导，该向导在前述的 DRA 中概述过。如果 DRA 当前没有检测到故障，则 Advise 和 Recover 按钮将是不可用的。

13.8.2　用户直接恢复

如果没有使用 RDBMS 关于正确执行数据库恢复的建议，企业管理器仍然会提供一些机制，通过这些机制可以选择不同的恢复类型并通过一系列有指导性的步骤来执行恢复。通过 Availability 页面上的 Perform Recovery 链接，首先将看到 Oracle Advised Recovery(前面讨论过)，然后看到 User Directed Recovery 部分。

1. 整体数据库恢复

恢复的第一种类型是整体数据库恢复。在 Whole Database Recovery 标题的下方有一些选项：Recovery to Current Time or Previous Point-in-Time 选项、Restore All Datafiles 选项，以及 Recover from Previously Restored Datafiles 选项。

Recover to Current or Previous Point in Time 选项

第一个选项的作用是还原数据文件并使用归档日志恢复它们。在该选项中需要决定何时打开数据库。实际上，如果完成该向导，将会发现生成的 RMAN 脚本如下所示(单击 Edit RMAN Script 按钮以查看输出)：

```
run {
    restore database;
    recover database;
    }
```

另一个选项是执行时间点恢复(PITR)。我们可以指定需要恢复到的时间点以及日志序列号、SCN 或还原点。需要注意的是，当执行 PITR 并且打开 Flashback Database 时，就可以看到一个蓝色字体的提示让您考虑使用闪回数据库(在下一个屏幕中有选择使用闪回数据库的选项)。当指定时间(SCN 或日志)时，OEM 估计该时间点是否在闪回数据库可用的时间框架内，并且提供如下选项：Flashback 或 Traditional PITR。选择 Flashback 则打开 Flashback Wizard(闪回向导)；选择 Traditional PITR 则继续执行当前向导。第 15 章将更详细地介绍闪回数据库。

无论在 OEM 中执行何种类型的文件还原，都可以将还原文件的位置更改为新的位置。在 OEM 向导中，完全数据库恢复时唯一的选项是所有数据文件都在一个新的位置中，而不能为单个文件选择不同的文件位置。然而，当执行到该向导的末尾时，您总是可以选择 Edit RMAN Script 并在此处更改文件名。

Restore All Datafiles 选项

在 Perform Recovery 主页面中，第二个选项是 Restore All Datafiles。选择该选项可初始化没有同时执行恢复的数据文件还原。和前面的步骤一样，执行到该向导的末尾，单击 Edit RMAN Script 按钮可以显示如下代码：

```
run {
restore database;
}
```

如果选择在还原时重命名数据文件，则向导要求您为所有文件提供一个位置，并且显示复选框 Update the Control File to Use the Renamed Files。选择该复选框将修改当前控制文件以指向新位置的数据文件。这与执行 RMAN 命令 switch datafile all 的效果相同。上述操作只在实际还原数据库而不是在新的位置建立另一个数据文件集时执行。如果希望修改单个的文件名，可以在最后检查脚本。

如果选择在不恢复的情况下将数据文件还原到新的位置，OEM 仍然会关闭目标数据库，然后将其置于加载模式中。RMAN 实际上不需要这些操作，但 OEM 的智能只能执行到这一步，并且它将数据库还原视为恢复的第一步。无论如何，OEM 在文件还原后会保持数据库加载状态。我们必须进入到数据库主页并单击 Startup 按钮来启动数据库。

关于选项 Restore All Datafiles 还有最后一个注意事项。OEM 已经在该向导第一个页面的底部(在标题 Backup Validation 的下方)隐藏了 backup validate 命令。如果希望验证是否可以进行某种类型的还原(当前时间点或以前的时间点还原)，则可以选择该选项，然后按照前面的步骤执行向导。需要注意的是，如果单击 Edit RMAN Script 按钮，则生成的脚本有细微的区别：

```
run {
restore database validate;
}
```

Recovery from Previously Restored Datafiles 选项

最后，可以选择从以前还原的数据文件中恢复数据库。这意味着如果选择该选项，在 RMAN 中只会执行 recover 命令。这并不意味着如果选择将数据文件还原到不同的位置，就可以在该向导中指定这些文件。这只是恢复操作，它完全依赖于控制文件的 V$DATAFILE 列表来确定还原哪些文件。

位于底部的复选框可以让 OEM 在恢复后以只读模式打开数据库，从而可以验证该数据库。该选项可用来确认所指定时间点的正确性。

RMAN 实习：在 OEM 中执行数据库恢复

实习注意事项

在该实习中，通过使用在本章前面利用 OEM 获得的备份，执行 PITR 来恢复到过去的时间点。

步骤 1　进入到 Database | Availability | Perform Recovery 页面。选择 Recovery to the Current Time or a Previous Point-in-Time 选项，然后单击 Perform Whole Database Recovery 按钮，您将会被要求确认 EM 将数据库置于加载模式。

步骤 2　选择 Recover to a Prior Point-in-Time 选项。在该示例中选择对日期进行设置。单击 Next 按钮。

步骤 3　如果已经打开 Flashback Database 并且时间点在 Flashback 窗口内，则会显示一个屏幕，询问是否希望使用闪回数据库。选择 No, Use Traditional Point-in-Time Recovery 选项。单击 Next 按钮。

步骤 4　选择将文件还原到默认位置。单击 Next 按钮。这会显示 Schedule 页面，但是所有的恢复作业会立刻运行。单击 Next 按钮。

步骤 5　单击 Edit RMAN Script 按钮以查看 RMAN 运行的脚本。完成检查后单击 Cancel 按钮，然后单击 Submit Job 按钮。

步骤 6　OEM 显示表明作业已经成功提交的屏幕输出。单击 View Job 按钮。作业页面会显示每个作业步骤的历史记录，称为执行。可以刷新浏览器以获得最新的步骤和执行该步骤使用的时间。

步骤 7　单击 Name 列中的 Step:Recovery 链接，从而可以监控 RMAN 作业的执行过程。

2. 小范围恢复

在 User Directed Recovery 下的 Perform Recovery 页面上，可以在整个数据库或者在数据库中可恢复项的较小的子集上选择恢复的范围。除了整个数据库之外的任何对象。表空间、数据文件、归档日志、表和事务处理——即整个数据库之内的任何对象，都可以从该部分中执行恢复。

由此，既可以恢复逻辑对象，又可以恢复物理对象。一些逻辑对象，例如表空间，仍然需要物理介质还原/恢复操作。选择其他逻辑对象，例如表，就需要使用 Oracle 的闪回技术，该技术完全不同于介质恢复。物理对象非常直观：要恢复一个数据文件，只需要执行还原和恢复操作。也可以在该位置执行块介质恢复。注意，当从下拉列表中做出选择时，网页将刷新并显示与选择相关的子选项。

Oracle 已经在 OEM 中提供了完整的恢复选项集，因此由于篇幅的限制不可能在本章中基于每种类型的对象级恢复来演示在向导中的操作步骤。我们鼓励您执行每种不同类型的 OEM 恢复，从而熟悉 Oracle 在 OEM 中针对每种恢复类型使用的语言。通常，这种语言不同于 RMAN 中使用的语言。

13.9 备份管理和报告

实际上，OEM 在管理备份方面更能显示它的作用。从命令行中运行备份和恢复与从终端读取报告输出和列表命令是完全不同的事情。在 OEM 中，管理现存的备份更多的考虑用户的方便使用性。这种管理可划分为 3 种类型：管理当前备份、管理还原点，以及创建备份报告。

13.9.1 管理当前备份

在数据库的 Availability 选项卡上，单击 Manage Current Backups 链接可访问执行常见管理活动的页面：运行交叉检验，删除已不用的备份，删除到期的备份等活动的页面(如图 13-24 所示)。

图 13-24 Manage Current Backups 页面

在 Manage Current Backups 页面的顶部，可以看到一些常见任务的快速链接按钮：Catalog Additional Files、Crosscheck All、Delete All Obsolete，以及 Delete All Expired。这些按钮确切地执行相应的任务，第 14 章中将对此进行介绍。

在这些按钮下方的两个选项卡 Backup Sets 和 Image Copies 用于管理子集中的备份或单独的备份。Search 区域用于根据某个搜索向量查找备份。默认情况是列出所有备份。在 Results 区域中，可以通过单击对应的复选框来选择一个或多个备份，然后选择一个管理选项：Crosscheck、Change to Unavailable、Delete 或 Validate。

当选择其中一个选项时，Job Creation Wizard 可帮助调度维护作业。这就允许您安排特定的作业，例如交叉检验，在每隔一段固定的时间后就执行一次。也可以单击 Show RMAN Script 选项查看 RMAN 所做的工作。

13.9.2　管理还原点

单击 Maintenance 选项卡上的 Manage Restore Points 以打开新的页面，在该页面中可以创建新的还原点，查看已有还原点的信息，或者删除已有的还原点。在 OEM 中，可以查看还原点创建时间、所需存储空间的数量，以及创建 SCN。如果注意到 Guaranteed Restore Point 不可用(没有显示在下图中)，这是由于使用具有 NORMAL 权限的用户账户登录到目标数据库，但是需要用 SYSDBA 权限才可以创建有保证的还原点。此外，Recover Whole Database To 按钮只有在 SYSDBA 用户登录时才起作用。单击该按钮会要求您进行确认，然后打开 Perform Whole Database Recovery 页面。

13.9.3　创建备份报告

单击 Maintenance 选项卡上的 Backup Reports 链接可打开 Backup Reports 页面，该页面显示了数据库已知的所有备份作业。这些备份不仅是 OEM 作业获得的备份——OEM 实际上会查询恢复目录(或控制文件)以查找所有备份。Backup Reports 页面允许您根据状态、启动时间和类型进行搜索。

可以单击备份名称以检索关于特定备份的详细信息，如图 13-25 所示。这包括输入(备份的内容)和输出(创建的备份片)的详细概述。然后，我们得到如图 13-25 所示的详细清单：每个输入文件、它的检查点信息、压缩比率等。该清单中包含非常多的数据，这些数据很有价值，并且比 RMAN 命令行中的报告和列表选项更容易阅读和理解。

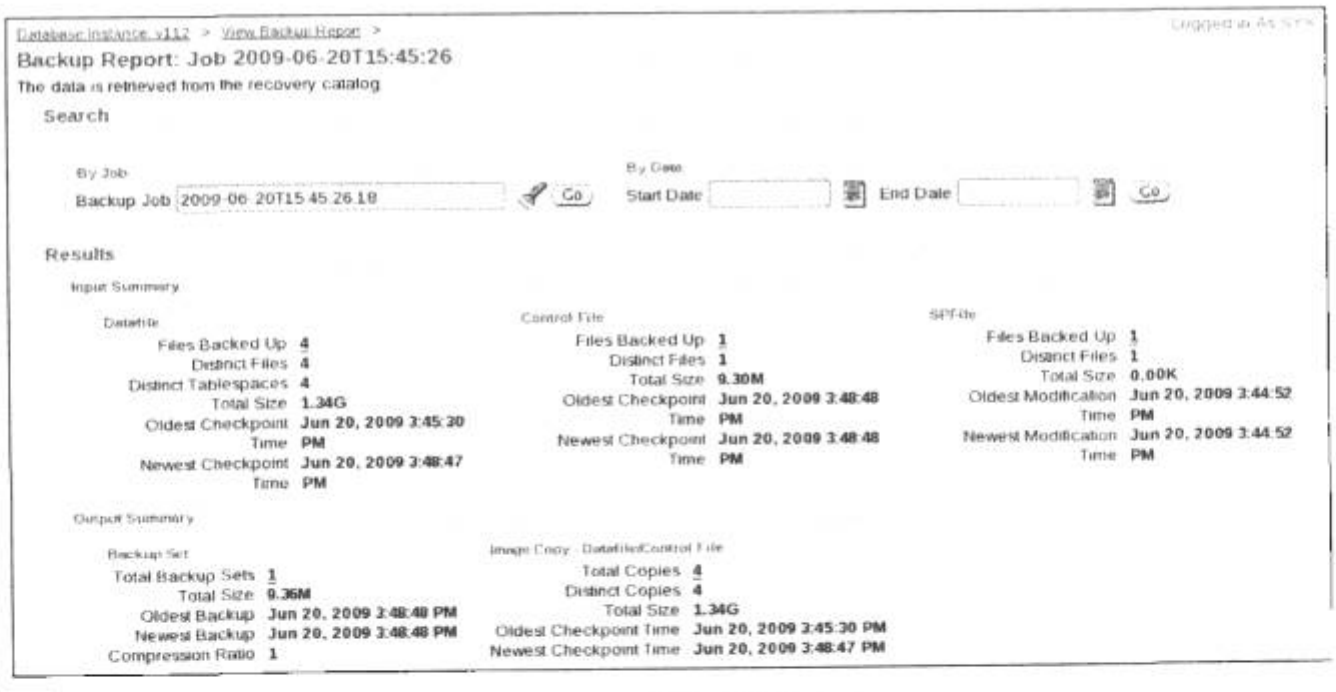


图 13-25　详细信息页面

13.10　在企业管理器中复制数据库

OEM 有一个执行数据库复制的子系统。OEM 的数据库复制过程利用 RMAN 功能来完成，但不要将其与使用 RMAN duplicate 命令的数据库复制(第 19 章将介绍这种复制方式)混淆。OEM 复制具有自己的逻辑和约束以区别于 RMAN 复制。相对于 10g 中的 OEM，

在 11.2 版本中，复制数据库与 RMAN 更相似。

OEM 可能在后台使用 RMAN，但数据库复制功能在逻辑上组织到完全不同的 OEM 区域中。我们需要从 EM 主页面进入到 Data Movement 选项卡，然后查看在 Move Database Files 下的 Clone Database 链接。只有在 Grid Control 中才可以复制到与目标数据库不同的服务器。Database Control 只可以复制到与目标数据库相同的服务器。

要做出的第一个决定就反映了在 EM 中进行复制的灵活性：进行复制可以选择用一个新的 RMAN 备份，使用策略空间(10g 类型复制)，或者使用现存的 RMAN 备份。最后一个选择是最直接、最有逻辑的，但在 10g 中是不可用的，所以，在本书中用它作为实习例子。

在选择使用现存的 RMAN 备份后，需要提供源主机凭证并单击 Continue 按钮。因为已经注册了带有已知的恢复目录的源数据库，所以不需要输入任何进一步的源数据库说明。只需要指定想要复制的时间点。

解决源数据库信息的问题后，接下来需要解决目标数据库信息的问题：Destination Oracle Home、Destination Host Credentials 和 Destination Database 的指定，以及并行度(使用 RMAN 信道的数目——在图 13-26 的说明中没有给出)。如果使用 Database Control，可以更改 Oracle Home，但是不可以更改主机。可以更改数据库名字，但它将默认一个不同于现存的源数据库的名字。在此，也可以更改新数据库从 File System 到 ASM 的存储类型，反之亦然。

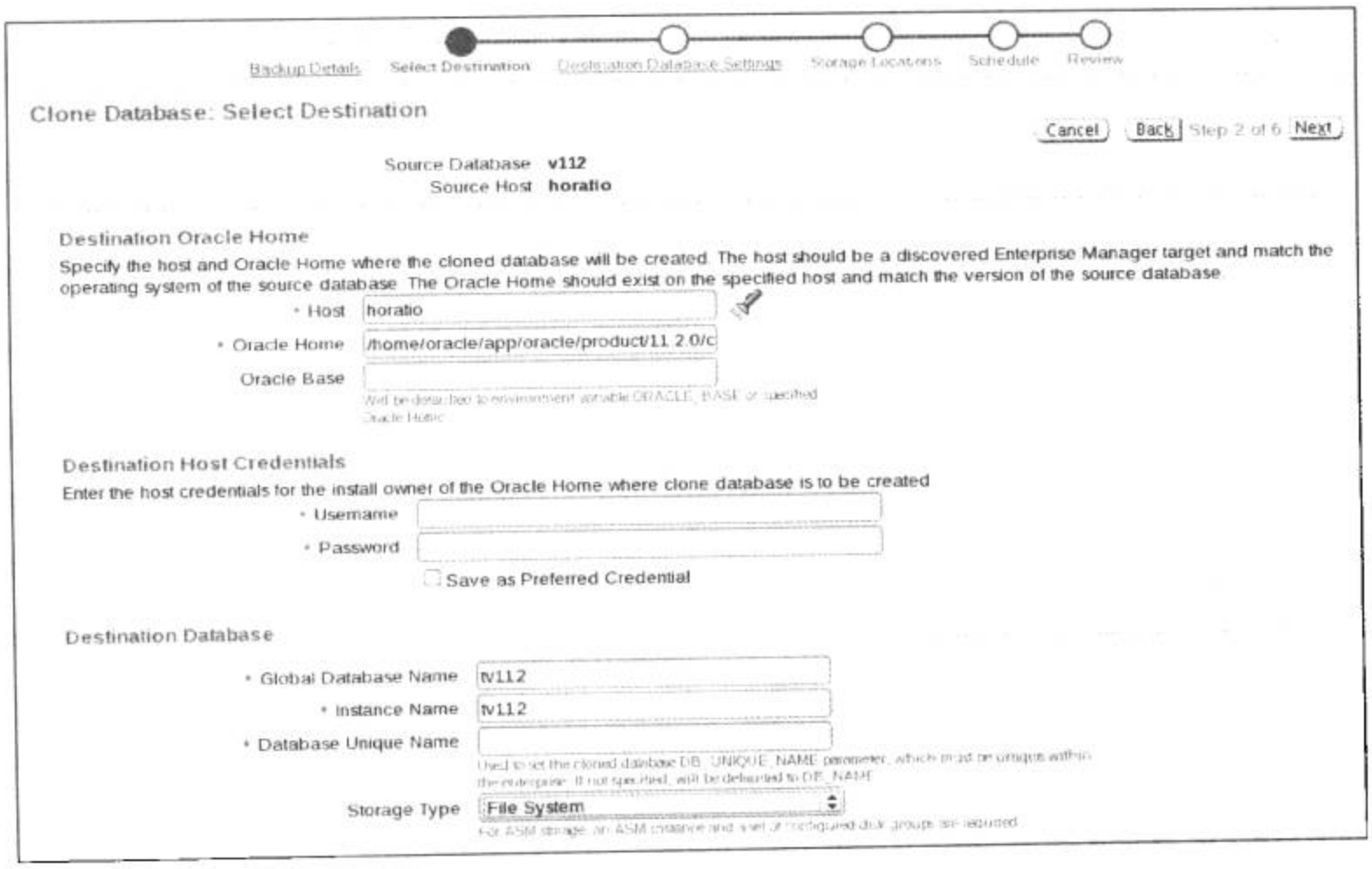


图 13-26 Select Destination 选项

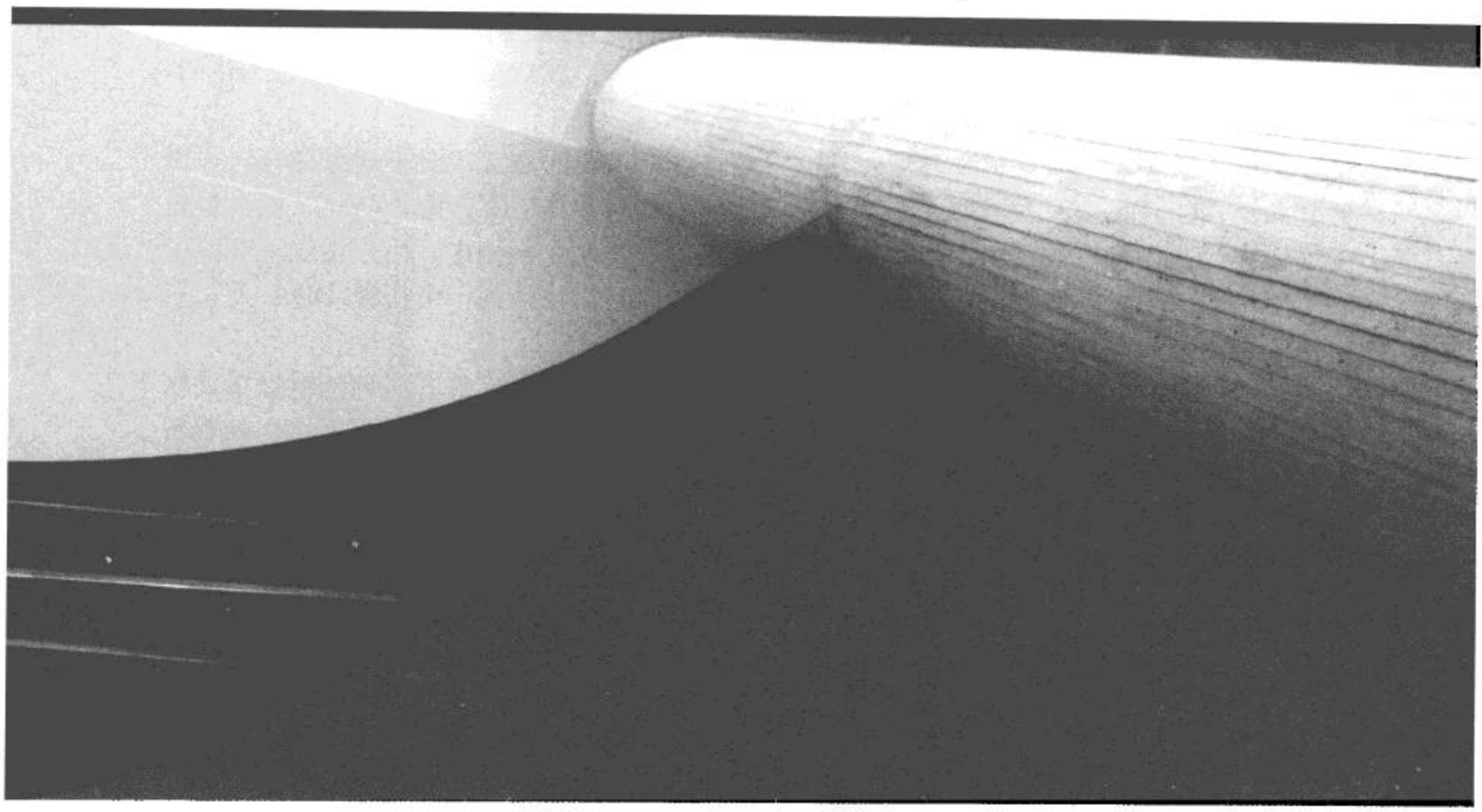
在向导的第三步中，指定目标数据库内部运作的元素，例如 Memory Confirmation。

您可以改变监听器并运行任何特定于您所在的环境的 Post Cloning 脚本。在下一页，指定文件位置，FRA 规范，归档日志目标文件，和存放诊断及审计文件的位置(存放 ADR 的位置)。然后，调度作业，审查规范，接着执行操作。

复制操作全部完成后，在目标服务器上将存在一个复制数据库，该数据库是开放的，可用的，并且 OEM 将发现该数据库，这意味着现在可以从 OEM 内部对该数据库进行管理——如果您使用的是 Grid Control。如果您使用的是使用 Database Control，则在新的复制中需要利用 emca 删除和重新建立 EM 存储库。这在复制过程中不会自动完成。

13.11　小结

本章讨论了 Oracle Enterprise Manager 11g Grid Control 和 Oracle Enterprise Manager 11g Database Control 的体系结构和配置。我们提供了一些示例以演示如何配置这些实用程序来获得 RMAN 备份以及如何配置数据库以实现在 OEM 中的恢复。我们也讨论了 Oracle 建议的备份策略，并且讲解了如何调度自定义的备份。接下来介绍了如何执行恢复。最后，我们展示了如何使用 OEM 来利用 RMAN 作为底层技术复制数据库。



第 14 章

RMAN 的高级恢复主题

本章将介绍一些 RMAN 的高级恢复主题。首先将介绍最难的也是最容易混淆的恢复主题：不完全恢复；然后再介绍其他的恢复主题。接着，介绍表空间时间点恢复以及如何验证备份。最后，还将介绍与 RMAN 相关联的 DBMS_BACKUP_RESTORE 数据包。

14.1 不完全恢复

顾名思义，不完全恢复是指不完全的数据库恢复。不完全恢复与完全恢复在许多方面是相同的，它们基本的命令集相同，但不完全恢复添加了一些其他命令。引起不完全恢复的原因有很多，如丢失了联机重做日志或归档的重做日志，或者出现重大的用户错

误。不完全恢复会影响整个数据库；换句话说，不能只对数据库的一部分执行不完全恢复操作，因为这会使数据库的一部分具有与这个数据库其余部分不同的系统更改号(SCN)或时间点。

不完全恢复影响的是整个数据库，这一点是非常重要的概念，对于初级 DBA 来说尤其如此。Oracle 要求数据库在启动时是一致的，如果数据库不一致，Oracle 就会报告出现错误。我们可以通过一个示例来说明这个概念，在这个示例中，用户错误地删除了表空间中的一个表，而他恰巧又没有备份这个表。此时，他找到一个初级 DBA 要求只将这个表空间恢复到该用户执行删除操作之前的时间点。

初级 DBA 起初可能会认为只需要还原损坏表空间的数据文件并将这些数据文件恢复到执行删除操作之前的时间点。这些操作看上去非常符合逻辑，实际上，这正是逻辑备份的操作。因此，初级 DBA 还原了数据文件，并将表空间恢复到删除操作之前的时间点，此时她的感觉非常良好。但是，这种感觉不会持续很久，因为当她试图打开数据库时，RMAN 会给出下面的消息：

```
ERROR at line 1:
ORA-01113: file 3 needs media recovery
ORA-01110: data file 3: 'D:\ORACLE\ORADATA\RECOVER\INDX01.DBF'
```

在这个示例中，Oracle 告诉用户恢复数据文件是成功的，但是这些数据文件恢复得不够，与数据库的其他部分不一致！因此，正如您所看到的，不完全恢复并非所有灾难的解决方案，它只是某些灾难的解决方案。如果需要某种形式的不一致恢复(将一些数据库数据还原至与数据库剩余部分不同的时间点)，可以查看本章后面讨论的表空间时间点恢复，或者在第 16 章中讨论的 Oracle 的闪回技术。

然而，如果需要将整个数据库还原到闪回数据库无法达到的过去的某个时间点，则需要使用不完全恢复。当执行数据库不完全恢复时，需要使用 resetlogs 命令，因为差不多所有的不完全恢复情况都要用到 resetlogs 命令。我们将在下面介绍该命令，随后还要讨论实际的不完全恢复方法，这些方法包括基于时间、SCN、日志序列或取消的恢复。

14.1.1 使用 resetlogs 命令

在不完全恢复期间，通常需要使用 resetlogs 命令打开数据库，这是因为我们要从已经建立的现有重做流中脱离出来，并且需要向 Oracle 说明这种情况。resetlogs 命令表示一个数据库逻辑生存期的结束和另一个数据库逻辑生存期的开始。数据库的逻辑生存期也称为一个对应物(incarnation)。每次使用 resetlogs 命令都会创建一个新的数据库对应物，这对于恢复操作来说非常重要，我们将在后面讨论这个问题。

每次使用 resetlogs 命令时，SCN 计数器不会被重置，不过 Oracle 会重置其他计数器(如日志序列号)，同时还会重置联机重做日志的内容(如果有必要还可以重新创建联机重做日志)。

从 Oracle Database 10g 开始，简化了通过 resetlogs 命令进行的恢复。在归档的重做日志名中添加了一个新的替换字符串(%r)，该字符串表示 resetlog ID 号。在 log_archive_dest_format 参数字符串中包括%r 时，归档的重做日志名在每个 resetlogs 命令中保持唯一。

这种改动以及其他的内部 Oracle 数据库改动使 Oracle 可以很容易地通过给定的 resetlogs 操作恢复数据库。因此，可以很容易地在执行 resetlogs 操作后立刻备份数据库。然而，我们仍然认为在任何不完全恢复后备份数据库是很好的想法。

注意：
在以前版本的 RMAN 中，需要重新创建所有的临时表空间临时文件。Oracle Database 10g 和以后的版本中则不需要这种操作，RMAN 在恢复数据库过程期间会创建这些数据文件，甚至在不完全恢复期间也是如此！

14.1.2　创建恢复点

使用 RMAN 执行不完全恢复操作时需要完成的一个工作是创建恢复目标。恢复目标是恢复进程的终点，通常我们基于一个时间点、一个指定的 SCN 或一个日志序列号来标识它。我们可以使用许多不同的方法创建恢复目标。首先，可以在 run 代码块中使用 set 命令与 until time、until SCN 或 until sequence 参数，下面的示例使用 set until time 命令创建了一个为 2006 年 7 月 1 日下午 3 点的恢复目标：

```
run
{
set until time "to_date('07/01/06 15:00:00','mm/dd/yy hh24:mi:ss')";
restore database;
recover database
alter database open resetlogs;
}
```

执行这条命令时，RMAN 会查找与恢复目标时间最近(并非恢复目标时间本身也不能是位于恢复目标之后的时间)的备份集，并且从这个备份集中还原数据库。如果数据库置于 NOARCHIVELOG 模式中，恢复操作会在备份集的时间点停止；否则在执行 recover 命令期间，Oracle 会在所定义的恢复目标(不包含恢复目标本身)上应用归档的重做日志(以及需要应用的任何增量备份)。

注意：
如果尝试恢复到特定备份的完成点，则必须恢复到备份集中文件的 CKP SCN 或 CKP TIME，在不同备份集的 RMAN 的 list 命令中列出了这些内容(例如 list backupset)。有时使用备份的 CKP TIME 并不够，还可能导致 ORA-1152 错误。

也可以选择在 restore 和 recover 命令中直接使用 until time、until SCN 或 until sequence 命令，这样就避免使用 run 代码块(我们倾向于使用这种方法)。下面是在还原和恢复数据库时所使用的 until time 命令示例：

```
-- We assume that your control file is intact
startup mount;
restore database UNTIL TIME
```

```
"TO_DATE('06/28/06 13:00:00','MM/DD/YY HH24:MI:SS')";
recover database UNTIL TIME
"TO_DATE('06/28/06 13:00:00','MM/DD/YY HH24:MI:SS')";
alter database open resetlogs;
```

14.1.3 基于时间的恢复

14.1.2 小节中给出了一些基于时间的恢复示例。这种恢复类型允许用户将数据库恢复到与指定时间一致的状态。当然，如果不存在能将数据库还原到用户请求的时间的有效备份或归档重做日志，Oracle 就会生成如下所示的错误：

```
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of restore command at 07/02/2006 21:33:02
RMAN-20207: UNTIL TIME or RECOVERY WINDOW is before RESETLOGS time
```

最后需要强调的是必须具备在我们所指定的恢复时间之前生成的数据库备份，此外还需要所有的归档重做日志。我们将在第 15 章中讨论 list 命令和 report 命令，这些命令允许用户确定 RMAN 还原数据库可以使用的备份。

14.1.4 基于 SCN 的恢复

我们在第 1 章中详细讨论过 SCN。Oracle 允许用户将数据库恢复到指定的 SCN。实际上，这并不是一种常用的恢复方法，不过最好了解一下它的用法。我们将在本章后面使用这种方法来恢复前一个数据库对应物。下面是一个将数据库还原到指定 SCN 的示例：

```
-- We assume that your control file is intact
startup mount;
restore database UNTIL SCN 10000;
recover database UNTIL SCN 10000;
alter database open resetlogs;
```

在这个示例中，我们将数据库还原到 SCN 10000(但是不包含这个 SCN)。

14.1.5 基于日志序列的恢复

RMAN 允许用户将数据库恢复到指定序列号的归档重做日志。如果归档的重做日志中存在间隙，使用这种恢复方法就非常方便。间隙通常意味着我们只能将数据库还原到间隙的开始点。下面是一个在 RMAN 中执行基于日志序列的恢复操作的示例：

```
-- We assume that your control file is intact
startup mount;
restore database UNTIL SEQUENCE 100 thread 1;
recover database UNTIL SEQUENCE 100 thread 1;
alter database open resetlogs;
```

在这个示例中，我们将数据库还原到日志序列为 100(但是不包含这个日志序列)的数

据库。

14.1.6　基于删除的恢复

RMAN 不支持类似于 SQL*Plus 所做的基于删除的恢复(虽然基于删除的恢复在一定程度上非常类似于基于日志序列的恢复)，而是希望用户了解恢复到的时间、SCN 或日志序列号，因为这样非常有意义。如果需要执行基于删除的恢复，则需要从 RMAN 中将数据文件还原到恢复时间点之前的一些时间点。我们需要手动析取基于删除的恢复所需的归档重做日志，然后需要从 SQL*Plus 客户端中执行数据库的基于删除的恢复。第 1 章介绍过执行基于删除的数据库手动恢复的过程。

14.1.7　使用还原点恢复

可以利用还原点定义数据库还原到的时间点。基于还原点还原数据库，只包含 restore 和 recover 命令的 to restore point 子句，如下所示：

```
-- Startup mount;
   Restore database until restore point TANGO_ONE;
   Recover database until restore point TANGO_ONE;
   Alter database open resetlogs;
```

也可以使用 run 代码块和 set until restore point 命令创建目标还原点，如下所示：

```
-- Startup mount;
run {
set restore point TANGO_ONE;
Restore database TANGO_ONE;
Recover database TANGO_ONE;  }
alter database open resetlogs;
```

14.2　其他 RMAN 恢复主题

我们已经介绍过许多使用 RMAN 的不同还原操作，不过在本节中还将讨论另外一些 RMAN 还原与恢复操作。首先将介绍如何从备份集中还原归档的重做日志；然后介绍从备份中还原控制文件副本；接着还要讨论还原数据文件副本和备份集备份；最后再讨论对备份的验证，这是 RMAN 中的一个重要功能。

14.2.1　只读表空间的恢复

在默认情况下，即使丢失了只读的数据文件，RMAN 也不会在执行完全数据库还原操作时还原只读的数据文件。要在完全恢复期间还原只读的数据文件，就必须在 restore 命令中使用 check readonly 参数，如下所示：

```
restore database check readonly;
```

需要注意的是，执行 recover tablespace 或 recover datafile 命令时，RMAN 的工作情

况是不一样的。使用上述任何一种 recover 命令时，不管表空间是否为只读状态都会执行恢复操作。

14.2.2 归档重做日志的还原

在使用 RMAN 的普通恢复过程中，不必恢复归档的重做日志。不过，偶尔也会要求还原一个或多个归档的重做日志。例如，我们可能需要使用 LogMiner 在备份中存储的归档重做日志文件里查找一些信息，在这种情况下，RMAN 允许用户使用 restore archivelog 命令还原指定的归档重做日志，如下所示：

```
restore archivelog all;
restore archivelog from logseq=20 thread=1;
restore archivelog from logseq=20 until logseq=30 thread=1;
```

我们还可能要使 Oracle 将归档的重做日志还原到默认位置以外的位置上，此时就需要使用 set 命令与 archivelog destination to 参数：

```
run
{
set archivelog destination to "d:\oracle\newarch";
restore archivelog all;
}
```

需要注意的是，set 命令没有替代方法，所以上面的示例要求使用 run 代码块。最后需要记住的是，如果 RMAN 认为一个归档的重做日志已存在，就不会在磁盘上还原这个归档的重做日志，即使设置的还原位置不同于默认的归档日志位置，Oracle 也不会在这个新的位置上恢复归档的重做日志。

14.2.3 数据文件副本的还原

我们可以从数据文件副本(而不是备份集)中还原数据库的数据文件。要实现这个功能，需要先使用 restore from datafilecopy 命令，然后再使用恢复数据库(或者表空间、数据文件)的 recover 命令，如下所示：

```
restore (datafile 5) from datafilecopy;
recover datafile 5;
sql "alter database datafile 5 online;"
```

需要注意的是，执行 restore 命令时，这条命令会识别需要还原的数据文件的最新副本，然后从这个副本中还原这些数据文件。数据文件的最新副本可能是在一个数据文件副本中，而不是在一个备份集中。在这种情况下，Oracle 会恢复这个数据文件副本。此外要注意的是，圆括号很重要，如果没有使用圆括号，restore 命令就会失败。

14.2.4 恢复损坏的数据块

即使与损坏数据块关联的数据文件一直联机，我们也可以通过使用块介质恢复(block

media recovery，简写为 BMR)执行块级恢复操作来修复 Oracle 数据库中的这些逻辑上的或物理上的损坏数据块。

怎样才能执行块介质恢复操作呢？正如接下来的示例所示，这是非常简单的。假设我们在查询一个 Oracle 表的时候接收到下面的错误消息：

```
ORA-01578: ORACLE data block corrupted (file # 19, block # 44)
ORA-01110: data file 19: 'd:\oracle\oradata\data\mydb_maintbs_01.dbf'
```

这个错误消息说明 MAINTBS 表空间中的一个数据块出现了损坏。当然，我们需要执行某些操作来修正这个错误。没有 BMR 时，我们必须从一个备份中恢复这个数据文件。在恢复的过程中，用户不能够使用该数据文件中的所有数据。

相反，可以使用 BMR 只恢复损坏的数据块，这时需要使用 blockrecover 命令。在上一个示例中，可以使用如下所示的 blockrecover 命令：

```
blockrecover datafile 19 block 44;
```

如果有必要，可以同时恢复多个数据文件中的多个数据块。下面是两个使用 blockrecover 命令执行这种操作的示例：

```
BLOCKRECOVER DATAFILE 19 BLOCK 44,66,127;
BLOCKRECOVER DATAFILE 19 BLOCK 44 DATAFILE 22 BLOCK 203;
```

Oracle 会跟踪在备份和复制期间发生的数据块损坏。如果检测到备份或复制操作出现损坏，由于 Oracle 不允许在备份中出现损坏，所以这个操作默认就会失败。当然，可以配置 RMAN 允许一定数量的损坏，但是一般不推荐这种用法。

我们可以使用 backup validate database 命令查看 RMAN 检测到的所有数据库损坏，这条命令会在 V$BACKUP_CORRUPTION 和 V$DATABASE_BLOCK_CORRUPTION 视图中填充检测到的所有损坏数据块。下面是 backup validate database 命令执行后的部分输出：

```
RMAN> backup validate database;
Starting backup at 18-JUL-09
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=117 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002 name=C:\ORACLE\ORADATA\BORG\SYSAUX01.DBF
input datafile file number=00003 name=C:\ORACLE\ORADATA\BORG\UNDOTBS01.DBF
input datafile file number=00001 name=C:\ORACLE\ORADATA\BORG\SYSTEM01.DBF
input datafile file number=00005 name=C:\ORACLE\ORADATA\BORG\EXAMPLE01.DBF
input datafile file number=00006 name=C:\ORACLE\ORADATA\BORG\UNDO_NEW_01.DBF
input datafile file number=00009 name=C:\ORACLE\ORADATA\BORG\SMALL_THREE01.DBF
input datafile file number=00004 name=C:\ORACLE\ORADATA\BORG\USERS01.DBF
input datafile file number=00007 name=C:\ORACLE\ORADATA\BORG\SMALL01.DBF
input datafile file number=00008 name=C:\ORACLE\ORADATA\BORG\SMALLTWO.DBF
```

```
channel ORA_DISK_1: backup set complete, elapsed time: 00:10:45
List of Datafiles
=================
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
---- ------ -------------- ------------ --------------- ----------
1    OK     0              12166        93440           4480511
  File Name: C:\ORACLE\ORADATA\BETA1\SYSTEM01.DBF
  Block Type Blocks Failing Blocks Processed
  ---------- -------------- ----------------
  Data       0              66011
  Index      0              12036
  Other      0              3227
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
List of Control File and SPFILE
===============================
File Type    Status Blocks Failing Blocks Examined
------------ ------ -------------- ---------------
SPFILE       OK     0              2
Control File OK     0              1426
Finished backup at 18-JUL-09
```

需要注意的是，示例输出中的所有数据文件被检测。一旦损坏检测完成，每个数据文件的具体报告被显示(在上面示例中，我们只是显示了数据文件 system01.dbf 中的报告)。同样需要注意的是，在 backup validate database 命令执行操作中，当前的控制文件和当前的 SPFILE 被检测。

如果损坏发生在复制操作期间，V$COPY_CORRUPTION 视图就会指明包含损坏的备份集。需要注意的是，V$BACKUP_CORRUPTION 是一个显示历史损坏的视图，V$DATABASE_BLOCK_CORRUPTION 则是一个显示当前数据块损坏的视图。一旦修正了数据库的块损坏，就需要重新运行 backup validate database 命令，然后查询 V$DATABASE_BLOCK_CORRUPTION 视图以确保不存在其他损坏。

我们已经提到：查询 V$DATABASE_BLOCK_CORRUPTION 视图可以查看损坏数据块的详细信息。如下所示，使用具有 corruption list restore 参数的 blockrecover 命令可以方便地修正 V$DATABASE_BLOCK_CORRUPTION 视图中的损坏数据块：

```
BLOCKRECOVER CORRUPTION LIST RESTORE UNTIL TIME 'SYSDATE - 5';
```

如果想从旧的 0 级还原并且验证 blockrecover 命令中块和重做流存在问题，我们可以使用 NOFILEUPDATE 测试运行，然后再应用和重做：

```
BLOCKRECOVER CORRUPTION LIST RESTORE UNTIL TIME 'SYSDATE - 5' NOFILEUPDATE;
```

在这个示例中，我们将还原损坏块列表中最近五天的所有损坏数据块。在上面的这

条命令中，还可以使用 *until time* 和 *until sequence* 关键字。

14.2.5　恢复前一个对应物

在前面讨论 resetlogs 命令时，我们介绍过一个数据库的对应物表示这个数据库的特定逻辑生存期。作为一名成功的 DBA，您会发现自己面临这样一种奇怪的还原情况，需要使用上次执行 resetlogs 命令打开数据库前生成的一个备份来还原数据库，或者可能需要还原到执行上一个 resetlogs 命令之前的时间点。本节将讨论如何使用恢复目录从前一个对应物中还原数据库(这是非常容易的任务)，以及不使用恢复目录的情况下如何从前一个对应物中还原数据库的方法(比较困难)；最后还将介绍一个非常独特的恢复案例，我们需要在这个案例中使用 resetlogs 命令来恢复数据库。

1. 使用恢复目录恢复前一个对应物

下面将介绍一个从前一个对应物中恢复数据库的示例。在这个示例中，我们假设使用恢复目录执行了备份操作，并且最近使用 resetlogs 命令执行过时间点恢复。现在，需要使用执行 resetlogs 命令时间点前的一个备份来恢复数据库。首先，需要使用 list incarnation 命令列出数据库的对应物(第 15 章将详细介绍这个命令)：

```
RMAN> list incarnation;
    List of Database Incarnations
    DB Key  Inc Key   DB Name  DB ID            CUR Reset SCN    Reset Time
    ------ --------- -------- ---------------- --- ----------- ----------
    1       2         RECOVER  2539725638       NO  763059      08-JUL-02
    1       123       RECOVER  2539725638       YES 764905      09-JUL-02
```

在这个列表中，我们会发现存在两个不同的数据库对应物：键为 2 的对应物和键为 123 的对应物。CUR 列说明我们正处于键为 123 的对应物(因为它在这列中为 YES)。因此，我们需要切换回数据库键为 2 的对应物。

既然已经知道要恢复哪一个对应物，就可以开始执行恢复操作(在这里，我们删除了一些输出以使示例更清楚)：

```
RMAN> startup force nomount
    Oracle instance started
    RMAN> reset database to incarnation 2;
    database reset to incarnation 2 in recovery catalog
    RMAN> restore controlfile;
    Finished restore at 10-JUL-02
    RMAN> restore database until scn 764904;
      Finished restore at 10-JUL-02
    RMAN> recover database until scn 764904;
    starting media recovery
    media recovery complete
    Finished recover at 10-JUL-02
    RMAN> alter database open resetlogs;
    database opened
```

```
new incarnation of database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
```

下面是在这个示例中所执行的步骤：

(1) 启动但不加载实例，这是因为我们先要得到一个与需要恢复数据库的对应物关联的控制文件。

(2) 使用 reset database to incarnation 命令为 RMAN 指示哪个对应物的备份集要用于恢复操作。

(3) 执行 restore controlfile 命令，使 RMAN 还原最新的控制文件。

(4) 加载数据库。

(5) 执行基于 SCN 的还原操作(本章前面部分讨论过这种操作)，还原这个数据库。我们要将数据库还原到最近一次执行 resetlogs 命令(它的 SCN 为 764905)之前的 SCN，因此执行 restore database 命令，限定将数据库还原到 SCN 764904。

(6) 执行 recover 命令，再次限定将数据库恢复到 SCN 764904，并等待恢复操作结束。

(7) 打开数据库，重置联机重做日志。

现在运行 list incarnation 命令时，执行结果如下所示：

```
RMAN> list incarnation;
List of Database Incarnations
DB Key  Inc Key    DB Name  DB ID       CUR  Reset SCN     Reset Time
------- --------- -------- ----------- ---- ------------- ----------
1       2          RECOVER  2539725638  NO   763059        08-JUL-02
1       123        RECOVER  2539725638  NO   764905        09-JUL-02
1       245        RECOVER  2539725638  YES  764905        10-JUL-02
```

可以看到，此时创建了数据库的新对应物(对应物键为 245)，还可以看到，为这个新对应物重置的 SCN 与对应物 123 的 SCN 相同。但是，重置时间(Reset Time)反映了创建对应物的实际时间。

2. 不使用恢复目录恢复前一个对应物

Oracle Database 10g 中简化了不使用恢复目录恢复前一个对应物的操作。现在，不需要复杂的过程就能非常简单地实现恢复！

为了通过前一个对应物进行恢复，需要一个包含前一个对应物信息的控制文件。在大多数情况下，这可能是当前的控制文件。如果当前的控制文件不了解需要恢复的对应物，则需要还原包含该信息的控制文件，从而使得能利用该方法恢复数据库。可以使用 list incarnation of database 命令查看控制文件了解哪个对应物，如下所示：

```
RMAN> list incarnation;
List of Database Incarnations
DB Key   Inc Key DB Name  DB ID         STATUS  Reset SCN    Reset Time
------- ------- -------- ------------ ------- ------------- ----------
4        4       ROB10R2  3753137102    PARENT  3599781      05-FEB-06
5        5       ROB10R2  3753137102    PARENT  3715262      08-FEB-06
6        6       ROB10R2  3753137102    CURRENT 4296046      20-FEB-06
```

在这个示例中，我们发现控制文件包含有关对应物(Inc Key)4、5 和 6 的信息。当前的对应物是对应物 6。如果需要还原对应物 3，则不可以使用这个控制文件，因为对应物 3 不在该控制文件中。

需要注意的是，没有连接恢复目录时的 list incarnation 输出与已连接恢复目录时的 list incarnation 输出有着一些细微的区别。这是因为信息是从控制文件中获得的，因此某些键(例如 Inc Key)将会不同。这很正常。如果希望复位到对应物 5，然后恢复数据库，则需要执行如下操作：

(1) 从 RMAN 中运行 list incarnation 命令，确定希望复位到哪个对应物。

(2) 关闭数据库。

(3) 启动加载数据库。

(4) 执行 reset database to incarnation 命令复位对应物。

(5) 使用 restore 命令还原数据库(例如 restore until sequence 100 thread 1)。

(6) 恢复数据库(recover until sequence 100 thread 1)。

(7) 使用 resetlogs 命令打开数据库。

如同我们所说的，就是这么简单！下面是复位对应物恢复的一些示例输出。当完成恢复时，应注意 list incarnation 命令中发生的变化。同样需要注意的是，如果 Oracle 需要通过 resetlogs 命令恢复对应物 6，它将会自动完成这项工作。我们已经删除了一些不需要的输出以防止产生误解。

```
RMAN> list incarnation of database;
    List of Database Incarnations
    DB Key  Inc Key  DB Name  DB ID            STATUS  Reset SCN  Reset Time
    ------- -------  -------- ---------------- ------  ---------- ----------
    1       1        ROB10R2  3753137102       PARENT  1          30-AUG-05
    2       2        ROB10R2  3753137102       PARENT  534907     03-OCT-05
    3       3        ROB10R2  3753137102       PARENT  3586765    04-FEB-06
    4       4        ROB10R2  3753137102       PARENT  3599781    05-FEB-06
    5       5        ROB10R2  3753137102       PARENT  3715262    08-FEB-06
    6       6        ROB10R2  3753137102       CURRENT 4296046    20-FEB-06
RMAN> shutdown immediate
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount
connected to target database (not started)
Oracle instance started
database mounted
RMAN> reset database to incarnation 5;
database reset to incarnation 5
-- Note the reset SCN on the current incarnation is greater than
-- the SCN we wish to restore to. This is why we had to do a
-- restore to the previous incarnation.
RMAN> restore database until scn 4296041;
Starting restore at 20-FEB-06
```

```
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=156 devtype=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: sid=155 devtype=DISK
... blah blah blah… more stuff here we don't really need to see...
RMAN> recover database until scn 4296041;
Starting recover at 20-FEB-06
using channel ORA_DISK_1
using channel ORA_DISK_2
... blah blah blah… more stuff here we don't really need to see...
RMAN> list incarnation;
List of Database Incarnations
DB Key  Inc Key  DB Name  DB ID        STATUS  Reset SCN  Reset Time
------- ------- -------- ------------ ------ ---------- ----------
1       1        ROB10R2  3753137102   PARENT   1         30-AUG-05
2       2        ROB10R2  3753137102   PARENT   534907    03-OCT-05
3        3       ROB10R2   3753137102  PARENT   3586765   04-FEB-06
4        4       ROB10R2  3753137102   PARENT   3599781   05-FEB-06
5        5       ROB10R2  3753137102   CURRENT  3715262   08-FEB-06
6       6        ROB10R2  3753137102   ORPHAN   4296046    20-FEB-06
RMAN> alter database open resetlogs;
database opened
RMAN> list incarnation;
List of Database Incarnations
DB Key  Inc Key  DB Name  DB ID          STATUS Reset SCN  Reset Time
------- ------- -------- -------------- ------ ---------- ----------
1       1        ROB10R2  3753137102     PARENT  1          30-AUG-05
2       2        ROB10R2  3753137102     PARENT  534907     03-OCT-05
3       3        ROB10R2  3753137102     PARENT  3586765    04-FEB-06
4       4        ROB10R2  3753137102     PARENT  3599781    05-FEB-06
5       5        ROB10R2  3753137102     PARENT  3715262    08-FEB-06
6       6        ROB10R2  3753137102     ORPHAN  4296046    20-FEB-06
7       7        ROB10R2  3753137102     CURRENT 4296046    20-FEB-06
```

14.3 表空间时间点恢复

表空间时间点恢复(TSPITR)是 RMAN 用于简化 Oracle Database 10g 中操作的另一个功能。以前关于创建辅助实例的许多争论现在已经平息，虽然仍然可以选择手动创建辅助示例。但为什么要使用表空间时间点恢复？考虑如下情况：用户错误地删除了 3 个表，此外还截断了几个表，此时他只有请求管理员帮助恢复这些表。幸运的是，RMAN 允许我们执行表空间时间点恢复。在很多情况下，使用 TSPITR 是非常方便的。这些情况包括意外地删除了一个表(如同本章的第一个示例所示)，或包含大量数据的程序被加载时由于损坏数据而逻辑中止。

为了学习如何执行 TSPITR，您应当掌握 Oracle Database 10g 与 TSPITR 相关的一些术语：

- **辅助目标(auxiliary destination)**：Oracle 将在其中创建辅助集的可选位置。
- **辅助实例(auxiliary instance)**：这是我们创建的临时实例，RMAN 使用这个实例执行 TSPITR。完成 TSPITR 后，可以删除辅助实例。
- **辅助集(auxiliary set)**：需要执行 TSPITR 的其他目标数据库文件集。辅助集包括备份控制文件、回滚和撤消段表空间数据文件、SYSTEM 表空间数据文件、辅助数据库的联机重做日志，以及一个可选的位于辅助数据库中的临时表空间。
- **恢复集(recovery set)**：要执行 TSPITR 的表空间/数据文件集。
- **目标实例(target instance)**：包含将要恢复的表空间。

在下一节中将查看使用 RMAN 执行自动表空间时间点恢复的示例。我们也将说明使用 RMAN 手动执行表空间时间点恢复的操作。

14.3.1　执行自动的 TSPITR

在该示例中，我们将让 Oracle 创建辅助实例。成功的 TSPITR 的基本步骤是：

(1) 为 TSPITR 做准备。

(2) 执行实际的 TSPITR。

下面将详细介绍这些步骤。我们也将介绍使用已经准备好的辅助实例执行自动的表空间时间点恢复。

1. 为 TSPITR 做准备

在开始执行 TSPITR 之前需要完成一些步骤，本节中就将讨论这些步骤：

- 确定还原的时间点；
- 确保对象完全包含在希望还原的表空间中；
- 保存可能丢失的对象或数据。

确定还原的时间点

这里最关键的因素是确定将表空间还原到的时间点。我们需要认真对待这项操作，因为如果没有使用恢复目录，则表空间的恢复是一次性的过程。如果错误地标识了恢复的时间点，则不能重新来过。如果使用恢复目录，则不存在这种限制。

确保传送集中的对象是自包含的

我们也应该使用 TS_PITR_CHECK 视图来确保恢复集是完整的，并且标识所有可能要用到的其他表空间。例如，假定我们有一个表空间 TEST_RECOVER,需要使用 TSPITR 还原它。我们首先需要检查 TS_PITR_CHECK 视图来确保没有其他相关的表空间。下面是检查 TEST_RECOVER 表空间是否可以单独传送的查询示例：

```
Set lines 132
Column obj1_owner format a20
column obj1_name format a20
column obj1_type format a20
column reason format a60
SELECT obj1_owner, obj1_name, obj1_type, reason
```

```
FROM SYS.TS_PITR_CHECK
WHERE ( TS1_NAME IN ('TEST_RECOVER')
       AND TS2_NAME NOT IN ('TEST_RECOVER') )
OR ( TS1_NAME NOT IN ('TEST_RECOVER')
       AND TS2_NAME IN ('TEST_RECOVER') );
```

如果没有冲突，则不会返回任何行。如果存在冲突，则会看到描述的每个冲突的行，如下所示：

```
OBJ1_OWNER   OBJ1_NAME       OBJ1_TYPE  REASON
------------ -------------- ---------- ------------------------------
SCOTT        TEST_TSPITR    TABLE      Tables and associated indexes
                                       not fully contained in the
                                       recovery set
```

在当前情况下，我们具有显示为在另一个表空间中创建的索引。该索引与 TEST_TSPITR 对象关联。我们需要查找该索引位于哪个表空间中，并且也还原该表空间。

保存可能在恢复过程中丢失的对象或数据

显而易见的是，如果将 TEST 表空间还原到下午 2 点，则该时间点之后的任何表空间更改(新建对象、更新/插入或删除操作)将会丢失。丢失这些对象可能没有问题，但假设我们需要保存这些数据，则需要导出将要保存的数据(或者将其复制到数据库中的其他位置)。Oracle 提供了视图 TS_PITR_OBJECTS_TO_BE_DROPPED，该视图列出了将在恢复操作期间丢失的所有对象。使用该视图可以确定表空间中的对象在恢复之后的状态。

例如，如果将 TEST_TSPITR 表空间还原到 2006 年 2 月 20 日 23:40 的时间点，则会丢失 TEST_TSPITR_TWO 对象，如同下面的示例输出所示。

```
SQL>select * from TS_PITR_OBJECTS_TO_BE_DROPPED where
 2> tablespace_name='TEST_RECOVER';

OWNER      NAME                 CREATION_TIME       TABLESPACE_NAME
---------- -------------------- ------------------- ---------------
SCOTT      TEST_TSPITR_TWO      02/20/2006 23:42:46 TEST_RECOVER
SCOTT      TEST_TSPITR          02/20/2006 23:26:26 TEST_RECOVER
SCOTT      TEST_TSPITR_THREE    02/21/2006 00:18:29 TEST_RECOVER
```

2. 执行实际的 TSPITR

Oracle Database 10*g* 将为我们执行自动的 TSPITR，这意味着它将创建辅助实例。在这种情况下，我们只需要连接目标数据库和可选的恢复目录(如果有的话)，并且执行 recover tablespace 命令。RMAN 将为我们完成剩余的工作。

在下面的代码段中，我们提供了使用 recover tablespace 命令恢复 TEST_RECOVER 表空间的示例。在该示例中，我们使用可选的参数 auxiliary destination 来指示 RMAN 和 Oracle 应该在何处创建与辅助数据库关联的文件。使用该参数使得该恢复成为一个具有自动化实例的自定义 TSPITR。如果没有使用该参数，TSPITR 就称为完全自动的 TSPITR 恢复。

需要注意的是，如果使用 auxiliary destination 参数，则应该已经创建目标目录，并且 Oracle 必须能够写入到该目标目录。同样需要注意的是，在目标路径名中没有后缀的斜杠(/或\，这取决于当前的 OS)，包括斜杠将会导致 TSPITR 失败(并且获得的错误消息无法确切地描述该问题)。下面是 recover tablespace 命令的示例，我们使用该命令在 Oracle Database 10*g* Release 2 中成功执行 TSPITR：

```
recover tablespace test_recover
   until time "to_date('02/21/2006 00:18:00','mm/dd/yyyy hh24:mi:ss')"
   auxiliary destination 'c:\oracle\auxdest';
```

注意：
为了执行自动的 TSPITR，必须事先在目标上配置信道。通过这种方式，用于辅助实例中的信道将与目标上的信道相同。

一旦完成 TSPITR，我们就应该能够查看已恢复表空间中的对象，并且发现它们已经被恢复到请求的时间点。我们需要将恢复的表空间改回到联机状态以使用它们。在 RMAN 中，可以执行如下命令：

```
Sql 'alter tablespace test_recover online';
```

如果产生错误，Oracle 就会保持辅助实例和相关数据文件的完整性。我们可以尝试修正该问题并重新启动恢复操作。在这种情况下，使用 auxiliary 参数重新启动 RMAN，并且连接辅助实例(14.3.2 小节中将演示该操作)。

如果辅助实例的创建没有完全成功，可以很容易地删除辅助实例和它的服务，而不是使用手动的 TSPITR 过程重新启动恢复操作。首先，指出失败的原因，然后删除辅助实例/服务，并且重新启动自动的 TSPITR 过程。当作为 SYSDBA 登录时，我们可以通过从 SQL *Plus 中执行如下命令来删除辅助实例/服务：

```
exec dbms_backup_restore.manageauxinstance('auxiliary_sid_name',1);
```

需要注意的是，我们需要用 Oracle 分配给辅助实例的 SID 替换 auxiliary_sid_name 占位符。SID 的名称将列在 RMAN 输出中。这使得在启动 TSPITR 恢复之前清除所有旧的辅助实例。可以在执行该命令后进入到辅助目标目录，并删除该目录中的任何文件。

3. 带有自动化实例的自定义自动 TSPITR

本章前面提及，在执行 TSPITR 时可以自定义自动化实例创建的一些方面。我们演示了使用 auxiliary destination 参数来指示应该在何处创建恢复集。

下面是一些用来自定义 TSPITR 的创建，并且仍然允许 Oracle 创建实例的其他方法：

- 使用 set newname 命令指示恢复集的单个数据文件的位置。
- 使用 configure auxname 命令定义辅助实例的名称。
- 为辅助实例创建自己的参数文件，并且在参数文件中提供参数(例如 db_file_name_convert)。通过在$ORACLE_HOME/rdbms/admin 中创建名为 parms_auxint.ora 的文件来完成这项工作(文件名和位置取决于当前的 OS)。作为选择，我们可以使

用 RMAN 命令 set auxiliary instance parameter file 来指示辅助实例参数文件驻留在客户端上的路径。查看下一小节以了解配置 TSPITR 的实例时应用的参数信息。

一旦已经自定义了辅助实例，就可以通过执行 recover tablespace 命令让 RMAN 创建该辅助实例，如同本小节中所介绍的。

14.3.2 手动执行 TSPITR

Oracle 支持为 TSPITR 创建自己的辅助实例。也可以使用手动的 TSPITR 来完成失败的自动 TSPITR 操作。首先，仍然要应用前面的准备阶段中列出的步骤，接着必须准备辅助实例，然后执行 TSPITR 过程。我们将在下面几节中介绍这些步骤。

1. 准备辅助实例

首先，需要启动和运行辅助实例。前面已经定义过，辅助实例只是 RMAN 用于执行 TSPITR 的临时实例。辅助实例必须驻留在与目标数据库相同的计算机中，并且在辅助实例上不能执行任何类型的 DML。

在启动 TSPITR 前，需要准备辅助实例。这是一个不需要 RMAN 参与的手动进程，但是又和创建普通的数据库实例没有什么不同。执行下面的步骤可以创建辅助实例：

(1) 创建口令文件。

(2) 如果从 OEM 中运行还原，则创建数据库参数文件。

(3) 如果在 Windows NT 上运行 Oracle，则使用 oradim 程序添加数据库服务(例如 oradim –new –sid recover)。

(4) 启动辅助实例，并检查是否能够连接网络。

您应当大致了解这些步骤，这些步骤都是典型的 DBA 任务。本质上，这与执行 create database 命令创建数据库实例非常相似。不过，我们还将简单介绍一些与 TSPITR 有关的问题。

配置辅助参数文件

辅助数据库的参数文件不同于目标数据库所使用的参数文件。我们使用与配置普通数据库参数文件大致相同的方法来配置用于辅助实例的参数文件。在辅助实例的参数文件中应该包含一些附加的参数，表 14-1 列出了这些参数。

表 14-1 辅助实例参数文件的一些附加参数

参 数 名	可选或必要	说 明
db_name	可选	与目标数据库相同的名称
lock_name_space	必要	不同于创建辅助数据库所在系统上其他数据库名的唯一名称。在示例中为 aux1
db_file_name_convert	可选	用 RMAN 还原辅助数据库中的数据文件时，该参数用于为这些数据文件定义一组文件命名转换模式。它可以代替 RMAN 的 configure auxname 命令

(续表)

参　数　名	可选或必要	说　　明
log_file_name_convert	可选	用 RMAN 还原辅助数据库中的重做日志文件时，该参数用于为这些重做日志文件定义一组文件命名转换模式。它可以代替 RMAN 的 set newname 命令
control_files	必要	这个参数定义了辅助实例控制文件的名称和位置。辅助实例控制文件的名称应当不同于所定义位置中已有的其他控制文件名
remote_login_passwordfile	可选/必要	用于允许 RMAN 通过 Oracle Networking 服务连接辅助数据库。要求存在一个当前口令文件。如果要本地连接辅助数据库，就不需要设置该参数
Compatible	必要	必须与目标数据库的设置相同
db_block_size	可选/必要	如果在目标数据库上设置了该参数，在辅助数据库中也必须设置为同样的参数值

如果您需要了解配置这些参数的更多信息，可以参阅 Oracle 的参考手册。在下面的示例中，我们只简单地复制目标实例的参数文件并做一些修改。下面是复制并修改后的辅助数据库实例的参数文件：

```
# These parameters already exist in the target database parameter file and
# require no changes.
db_name=rob10r2
db_block_size=8192
db_cache_size=8388608
timed_statistics=TRUE
shared_pool_size=110M
large_pool_size=1M

# Note that remote_loging_passwordfile must be set to exclusive
# if we want to connect to the aux database through RMAN via Oracle Net.
# remote_login_passwordfile=EXCLUSIVE
compatible=10.2.0.1

# These parameters required changes to directory locations
# to reflect the new database instance.
background_dump_dest= c:\oracle\auxdest
core_dump_dest= c:\oracle\auxdest
user_dump_dest= c:\oracle\auxdest
control_files=c:\oracle\auxdest\control01.ctl

# These are new parameters that are exclusive to the aux database
db_unique_name=recover
db_create_file_dest=c:\oracle\auxdest
log_file_name_convert=
```

```
('C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2','d:\oracle\auxdest')
```

注意：
我们在这儿将 c:\oracle\auxdest 中的所有内容转储到参数文件中。执行该操作是由于辅助数据库具有临时特性。当然，可以根据个人喜好创建较为复杂的目录结构。

启动辅助实例和检查网络连通性

准备好辅助实例后，就可以启动这个辅助实例。在下面的示例中，可以使用 startup nomount 命令启动辅助实例：

```
sqlplus "sys as sysdba"
startup nomount pfile=c:\oracle\auxdest\initrecover.ora
```

启动辅助实例后，使用 Oracle Net 来连接这个辅助实例，以确认网络连通性是正确的。

执行带有手动辅助实例的 TSPITR

要执行实际的恢复操作，就必须连接 RMAN。由于要同时连接目标数据库、辅助数据库和恢复目录(如果要使用恢复目录)，所以要使用的连接字符串稍有不同。在下面的示例中，可以使用 auxiliary 命令行参数与 target、catalog 参数来连接辅助数据库：

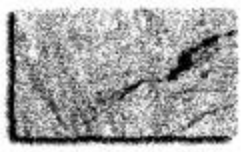
```
C:\Oracle\auxdest>rman target=sys/robert@rob10r2 auxiliary=/
Recovery Manager:Release 10.2.0.1.0 Production on Thu Feb 23 10:25:49 2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.
connected to target database: ROB10R2 (DBID=3753137102)
connected to auxiliary database: RECOVER (not mounted)
```

现在已经连接了 RMAN，我们需要能够加载辅助(副本)数据库。为了完成这项工作，我们还原一个副本控制文件，然后打开辅助数据库。需要注意的是，alter database archivelog current 命令在这里非常重要，它确保用于副本数据库恢复的所有重做都可用。下面是使用 restore clone controlfile 命令还原控制文件的示例，然后是加载副本数据库的示例。需要注意的是，这些命令包含在 run 代码块中。另外注意 sql clone 命令指示应该在副本数据库上执行 SQL 命令，与在目标数据库上执行的不带 clone 参数的 sql 命令相对应。

```
Run {
Set until time "to_date(' 02/23/2006 11:42:55','mm/dd/yyyy hh24:mi:ss')";
Restore clone controlfile;
sql clone 'alter database mount clone database';
sql 'alter system archive log current';
}
```

现在我们需要为 TSPITR 做准备。首先，我们需要知道还原的数据文件。我们需要与如下表空间相关的数据文件。

- **SYSTEM 表空间数据文件。**SYSTEM 表空间数据文件的数据文件号至少是 1(当然，如果向 SYSTEM 添加数据文件，则可能有其他数据文件号)。
- **UNDO 表空间数据文件。**
- **临时表空间(TEMP)数据文件。**如果当前没有这些表空间，也可以手动地创建它们。
- 将要还原的表空间数据文件(在当前情况下是 TEST_RESTORE)。



注意：
在当前情况下不需要还原 SYSAUX 表空间。

因此，接下来使用 SQL *Plus 在数据库中查找这些位置。我们使用 DBA_DATA_FILE 视图来查找与这些表空间数据文件关联的 FILE_ID。也要给出实际执行 TSPITR 的表空间的文件名。在当前示例中，下面是具体的输出(当然，您的输出可能会有所不同)：

```
SQL> select tablespace_name, file_id
  2> from dba_data_files
  3> where tablespace_name in
  4> ('SYSTEM','UNDOTBS1','TEST_RECOVER');

TABLESPACE_NAME                  FILE_ID
------------------------------ ----------
UNDOTBS1                                2
SYSTEM                                  1
TEST_RECOVER                            7

SQL> select file_name from dba_data_files where file_id=7;

FILE_NAME
--------------------------------------------------------------------------------
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\TEST_RECOVER_01.DBF

SQL> select tablespace_name, file_id
2> from dba_data_files
TABLESPACE_NAME                  FILE_ID
------------------------------ ----------
TEMP                                    1
```

在开始创建辅助数据库之前，我们需要将执行恢复的表空间的脱机。对于每个表空间，执行 SQL 命令 alter tablespace tablespace_name offline for recover;如同下面的示例所示：

```
Sql 'Alter tablespace test_recover offline for recover';
```

我们已经准备好实际地创建用于恢复的辅助(副本)数据库。首先，我们再次使用 until time 参数，因为这是新的 run 代码块。通过这种方式可以确保获得正确的数据文件。接下来，使用 set newname for clone 命令来确保在还原期间正确地命名这些文件。在该示例中使用了 OMF，因此可以很容易地使用 new 参数。我们也对数据文件 7 执行了 set newname

命令，为还原该数据文件做好准备，然后执行 switch clone tempfile all 命令来重命名临时文件。最后，还原数据文件并运行副本数据库：

```
Run {
    Set until time "to_date('02/23/2006 11:42:55','mm/dd/yyyy hh24:mi:ss')";
    Set newname for clone datafile 1 to new;
    Set newname for clone datafile 2 to new;
    set newname for clone tempfile 1 to new;
    Set newname for datafile 7 to
    "C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\TEST_RECOVER_01.DBF";
    Switch clone tempfile all;
    restore clone datafile 1, 2, 7;
    switch clone datafile all;
    sql clone "alter database datafile 1 online";
    sql clone "alter database datafile 2 online";
    sql clone "alter database datafile 7 online";
    sql clone "alter database mount clone database";
    }
```

现在打开副本数据库，准备执行 TSPITR：

```
Run {
    Set until time "to_date('02/23/2006 11:42:55','mm/dd/yyyy hh24:mi:ss')";
    recover clone database
    tablespace "TEST_RECOVER", "SYSTEM", "UNDOTBS1" delete archivelog;
    alter clone database open resetlogs;
    }
```

现在准备执行恢复过程中的最后一步。我们需要将元数据从副本数据库导出到主机数据库。下面是该过程：

```
C:\Oracle\auxdest>exp 'sys/robert as sysdba' point_in_time_recover=y
    tablespaces=test_recover file=tspitr_a.dmp

    Export: Release 10.2.0.1.0 - Production on Thu Feb 23 12:44:30 2006
    Copyright (c) 1982, 2005, Oracle. All rights reserved.
    Connected to: Oracle Database 10g Enterprise Edition
    Release 10.2.0.1.0 - Production
    With the Partitioning, OLAP and Data Mining options
    Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character
    set
    Note: table data (rows) will not be exported

    About to export Tablespace Point-in-time Recovery objects...
    For tablespace TEST_RECOVER ...
    . exporting cluster definitions
    . exporting table definitions
    . . exporting table                          TEST_TSPITR
    . . exporting table                      TEST_TSPITR_TWO
    . . exporting table                    TEST_TSPITR_THREE
```

```
. exporting referential integrity constraints
. exporting triggers
. end point-in-time recovery
Export terminated successfully without warnings.
```

现在关闭副本数据库：

```
C:\Oracle\auxdest>sqlplus "/ as sysdba"
SQL*Plus: Release 10.2.0.1.0 - Production on Thu Feb 23 12:47:05 2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.
Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

最后，将元数据导入到源数据库。需要注意的是，前面已经将数据文件还原到源数据库，因此不需要物理移动该文件。

```
C:\Oracle\auxdest>imp 'sys/robert as sysdba'
point_in_time_recover=y file=tspitr_a.dmp
```

现在的剩余工作就是将表空间联机：

```
Alter tablespace test_recover online;
```

2. 在手动执行 TSPITR 后执行的操作

完成 TSPITR 后，需要执行其他一些操作。首先，应当重新连接 RMAN 与目标数据库(如果使用了恢复目录，还要连接 RMAN 与恢复目录)，并且备份刚恢复的表空间。我们还要删除多余的辅助/副本实例。

14.3.3　TSPITR 限制

TSPITR 具有许多限制，下面概述了这些限制：

- 不能还原包含 SYS 用户特有对象的表空间。
- 不能用 TSPITR 恢复含有复制主表的表空间。
- 不支持使用快照日志的表空间。
- 不能还原含有回滚段的表空间。
- 如果要恢复的表空间中的对象是下面的任一类型，就不支持 TSPITR：
  - VARRAY
  - 嵌套表
  - 外部文件

此外，TSPITR 不能用于恢复删除的表空间。我们也不能恢复旧的对象统计表。如果使用不含恢复目录的 RMAN，还会存在下面这些与 TSPITR 有关的限制。

- 在要恢复的点和当前的点之间，目标数据库中撤消和回滚段的当前物理结构必须没有变化，也就是说，回滚段在恢复期间不能发生变化。
- 完成指定表空间的 TSPITR 后，这个表空间以前的所有备份就不能再用于以后的表空间的 TSPITR 恢复。这就是 TSPITR 后的表空间备份操作非常重要的原因，只有再次备份表空间后才能执行下一个 TSPITR。

14.4 验证备份可恢复

显然，如果备份不可恢复，它们就没有用处。RMAN 提供了一种不需要还原数据库就能检查数据库还原能力的方法，并且为用户提供了几个检查选项。本节将介绍一些验证数据库备份以及数据库可恢复性的不同方法。首先介绍 restore 命令的 verify 和 check logical 选项，然后介绍 validate backupset 命令。

14.4.1 restore preview 命令

如果希望查看 RMAN 使用哪个备份集来执行特定的恢复，则可以使用 restore preview 命令。该命令将给出还原所需的备份集的详细信息来执行命令行所指示的还原。例如，我们可以通过 restore database preview 命令为一个完全还原确定需应用哪个备份集。在下面的示例中，我们查看 restore database preview 命令的结果。在输出中，我们可以看到 RMAN 将应用增量级别为 0 的备份，然后应用增量级别为 1 的备份(如同单个 List of Backup Sets 报告上的 LV 列中所示)。该输出中也显示了将应用的归档的重做日志。最后，我们还得到了一些对不完全还原有用的信息：

```
RMAN> restore database preview;
    Starting restore at 23-FEB-06
    using channel ORA_DISK_1
    using channel ORA_DISK_2

    List of Backup Sets
    ===================
    BS Key  Type LV Size        Device Type Elapsed Time Completion Time
    ------- ---- -- ---------- ----------- ------------ ---------------
    210     Incr 0  66.36M      DISK        00:01:23     23-FEB-06
            BP Key: 210   Status: AVAILABLE   Compressed: YES
    Tag: AG20060223T142518
            Piece Name:
    C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2\BACKUPSET\2006_02_23
    \O1_MF_NNND0_TAG20060223T142518_1ZW6KK4H_.BKP
      List of Datafiles in backup set 210
      File LV Type Ckp SCN     Ckp Time  Name
      ---- -- ---- ---------- --------- ----
      2    0  Incr 4471679     23-FEB-06
    C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\UNDOTBS01.DBF
      3    0  Incr 4471679     23-FEB-06
```

```
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSAUX01.DBF
  4    0  Incr 4471679     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\USERS01.DBF
  5    0  Incr 4471679     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\EXAMPLE01.DBF

BS Key  Type LV Size         Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
211     Incr 0  83.30M      DISK        00:01:45     23-FEB-06
        BP Key: 211   Status: AVAILABLE Compressed: YES
Tag: TAG20060223T142518
        Piece Name:
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2\BACKUPSET\2006_02_23
\O1_MF_NNND0_TAG20060223T142518_1ZW6KYN2_.BKP

  List of Datafiles in backup set 211
  File LV Type Ckp SCN     Ckp Time Name
  ---- -- ---- ---------- --------- ----
  1    0  Incr 4471684    23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSTEM01.DBF
  6    0  Incr 4471684    23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\NEWTBS01.DBF
  7    0  Incr 4471684    23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\TEST_RECOVER_01.DBF
  8    0  Incr 4471684    23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\TEST_RECOVER_TWO_01.DBF

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
216     Incr 1 728.00K     DISK        00:00:38     23-FEB-06
        BP Key: 216 Status: AVAILABLE Compressed: YES
Tag: TAG20060223T144904
        Piece Name:
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2\BACKUPSET\2006_02_23
\O1_MF_NNND1_TAG20060223T144904_1ZW7Y40G_.BKP
List of Datafiles in backup set 216
  File LV Type Ckp SCN     Ckp Time Name
  ---- -- ---- ---------- --------- ----
  2    1  Incr 4472638     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\UNDOTBS01.DBF
  3    1  Incr 4472638     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSAUX01.DBF
  4    1  Incr 4472638     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\USERS01.DBF
  5    1  Incr 4472638     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\EXAMPLE01.DBF

BS Key  Type LV Size         Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
```

```
217     Incr 1  328.00K     DISK        00:01:10      23-FEB-06
        BP Key: 217 Status: AVAILABLE Compressed: YES
Tag: TAG20060223T144904
        Piece Name:
C:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\ROB10R2\BACKUPSET\2006_02_23
\O1_MF_NNND1_TAG20060223T144904_1ZW7Z98D_.BKP
  List of Datafiles in backup set 217
  File LV Type Ckp SCN     Ckp Time Name
  ---- -- ---- ---------- --------- ----
  1    1  Incr 4472653     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\SYSTEM01.DBF
  6    1  Incr 4472653     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\NEWTBS01.DBF
  7    1  Incr 4472653     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\TEST_RECOVER_01.DBF
  8    1  Incr 4472653     23-FEB-06
C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2\TEST_RECOVER_TWO_01.DBF

List of Archived Log Copies
Key   Thrd Seq      S Low Time  Name
----- ---- ------ - --------- ----
300   1    23       A 23-FEB-06 C:\ARCHIVE\ROB10R2ARC00023_0582936761.001

Media recovery start SCN is 4472638
Recovery must be done beyond SCN 4472653 to clear data files fuzziness

Finished restore at 23-FEB-06
```

一些供应商支持“再次调用”DR 中的介质，将其用于本地还原。restore database preview recall 支持这种功能，它允许您初始化一个再次调用操作以从远程灾难恢复站点中获取所需的备份文件，从而执行还原预览。

14.4.2 使用 validate 和 check logical 参数的 restore 命令

restore 命令提供一些验证数据库可恢复和备份有效的选项。首先，可以使用 backup 命令的 validate 参数使 RMAN 检查备份集，以确认数据库是可恢复的。使用 validate 选项时，Oracle 会检查用于恢复数据库的最新备份集以确认这个备份集是完整的，还会检查恢复操作需要使用的所有数据文件副本和归档的重做日志备份集，并确认它们也是完整的。此外，validate 选项会对备份集执行全面的验证操作，以确保这些备份集的完整性。验证操作的时间不会太长，并且这是确保数据库可恢复的一种方法。下面是一个对数据库执行 validate 操作的示例：

```
RMAN> restore database validate;
Starting restore at 05-JUL-02
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting validation of datafile backupset
```

```
channel ORA_DISK_2: starting validation of datafile backupset
channel ORA_DISK_1: restored backup piece 1
piece handle=D:\BACKUP\RECOVER\BACKUP_4QDSM5IB_1_1
tag=TAG20020703T221224 params=NULL
channel ORA_DISK_1: yalidation complete
channel ORA_DISK_2: restored backup piece 1
piece handle=D:\BACKUP\RECOVER\BACKUP_4RDSM5IC_1_1
tag=TAG20020703T221224 params=NULL
channel ORA_DISK_2: validation complete
Finished restore at 05-JUL-02
```

此外，使用 restore 命令的 check logical 参数能够更完全地检查最新备份集。这个命令使 RMAN 检查数据库的备份，如果这些备份通过物理损坏检查，还要检查备份的数据和索引段中是否存在逻辑损坏。如果发现逻辑错误，Oracle 会使用下面的一种方法来响应错误：

- 如果设置了 maxcorrupt 参数，并且还原检查逻辑操作期间的损坏数没有超过这个参数值，RMAN 就会在 Oracle V$表的 V$DATABASE_BLOCK_CORRUPTION 视图中填充损坏的数据块列表。
- 如果还原检查逻辑操作期间的损坏数超过了 maxcorrupt 参数值，操作就会终止。

在默认情况下，maxcorrupt 参数被设置为 0，因此任何逻辑损坏都会导致检查操作的失败。通过 set 命令可以修改默认的 maxcorrupt 参数值，但只能在 run 代码块的范围内使用这条命令。此外，maxcorrupt 参数是为每个数据文件单独设置的，而不是为所有的数据文件设置。下面的示例设置了允许出现一定损坏数的 maxcorrupt 参数，然后逻辑验证数据库的备份，在这个示例中，我们为数据库中的所有数据文件(1～5)都设置了 maxcorrupt 参数，同时不仅要检查出现的可恢复的最新备份集，还要查找备份集中的逻辑损坏：

```
RMAN> run {
2> set maxcorrupt for datafile 1,2,3,4,5,6 to 5;
3> restore database check logical validate;
4> }
executing command: SET MAX CORRUPT
Starting restore at 05-JUL-02
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting validation of datafile backupset
channel ORA_DISK_2: starting validation of datafile backupset
channel ORA_DISK_1: restored backup piece 1
piece handle=D:\BACKUP\RECOVER\BACKUP_4QDSM5IB_1_1
tag=TAG20020703T221224 params=NULL
channel ORA_DISK_1: validation complete
channel ORA_DISK_2: restored backup piece 1
piece handle=D:\BACKUP\RECOVER\BACKUP_4RDSM5IC_1_1
tag=TAG20020703T221224 params=NULL
channel ORA_DISK_2: validation complete
Finished restore at 05-JUL-02
```

14.4.3 使用 validate backupset 命令

使用具有 validate 参数或 check logical 参数的 restore 命令只能检查最新的备份集。不过，我们经常要检查任意的备份集，此时可以使用 validate backupset 命令。要使用这个命令，就必须首先确定要备份的备份集键(backup set key)。每个备份集在生成时都会被指派一个称为备份集键的唯一标识符。如下所示，使用 list backupset 命令可以确定为需要检查的备份集所分配的键值：

```
RMAN> list backupset;
    List of Backup Sets
    ===================
    BS Key  Type LV Size       Device Type Elapsed Time Completion Time
    ------- ---- -- ---------- ----------- ------------ ---------------
    141     Full    320K       DISK        00:02:09     03-JUL-02
            BP Key: 141 Status: AVAILABLE Tag: TAG20020703T221224
            Piece Name: D:\BACKUP\RECOVER\BACKUP_4QDSM5IB_1_1
      List of Datafiles in backup set 141
      File LV Type Ckp SCN    Ckp Time Name
      ---- -- ---- -------- --------- ----
      2       Full 647435   03-JUL-02 D:\ORACLE\ORADATA\RECOVER\REVDATA.DBF
      4       Full 647435   03-JUL-02 D:\ORACLE\ORADATA\RECOVER\TOOLS01.DBF
      6       Full 647435   03-JUL-02 D:\ORACLE\ORADATA\RECOVER\REVINDEX.DBF

    BS Key  Type LV Size       Device Type Elapsed Time Completion Time
    ------- ---- -- ---------- ----------- ------------ ---------------
    142     Full    113M       DISK        00:03:28     03-JUL-02
            BP Key: 142 Status: AVAILABLE Tag: TAG20020703T221224
            Piece Name: D:\BACKUP\RECOVER\BACKUP_4RDSM5IC_1_1
    List of Datafiles in backup set 142
    File LV Type Ckp SCN    Ckp Time Name
    ---- -- ---- -------- --------- ----
    1       Full 647439   03-JUL-02 D:\ORACLE\ORADATA\RECOVER\SYSTEM01.DBF
    3       Full 647439   03-JUL-02 D:\ORACLE\ORADATA\RECOVER\INDX01.DBF
    5       Full 647439   03-JUL-02 D:\ORACLE\ORADATA\RECOVER\USERS01.DBF
```

在这里，需要查看的是报告中的 BS Key 列，该列列出了备份集键号。需要注意的是，报告中也列出了备份集中文件的备份日期和时间。所有这些信息能够方便地标识要验证的备份集。确定需要检查的备份集后，如下所示，就可以运行 validate backupset 命令来验证这个备份集：

```
RMAN> validate backupset 141;
    using channel ORA_DISK_1
    using channel ORA_DISK_2
    channel ORA_DISK_1: starting validation of datafile backupset
    channel ORA_DISK_1: restored backup piece 1
    piece handle=D:\BACKUP\RECOVER\BACKUP_4QDSM5IB_1_1
    tag=TAG20020703T221224 params=NULL
    channel ORA_DISK_1: validation complete
```

```
RMAN> validate backupset 141 check logical;
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting validation of datafile backupset
channel ORA_DISK_1: restored backup piece 1
piece handle=D:\BACKUP\RECOVER\BACKUP_4QDSM5IB_1_1
tag=TAG20020703T221224 params=NULL
channel ORA_DISK_1: validation complete
```

14.5　跨平台的数据库移动和 RMAN

Oracle 数据库支持手动地跨平台移动数据库，即使这些平台具有不同的尾数格式(endian format)。尾数格式与字节排序有关，它有两种不同的格式，即大尾数和小尾数。如果要在不同尾数字节格式的平台之间移动数据库，则需要手动操作，并且使用 RMAN 的 convert datafile 或 convert tablespace 命令来将传送的数据文件转换为正确的尾数格式。

在本节中，我们将简要介绍可跨平台传送的表空间，然后讨论不同的尾数字节排序格式，接下来讨论转换表空间以传送到具有不同尾数字节格式的平台，最后介绍使用 RMAN 在具有相同尾数字节格式的平台之间移动。

14.5.1　可跨平台传送的表空间

DBA 有时会需要能够在开发部门的 NT Oracle 数据库和生产部门的 Sun Oracle 数据库之间移动表空间，有时则会需要在 Sun 平台和 AIX 平台之间移动表空间。从 Oracle Database 10g 开始，提供了该类问题的解决方案。

现在，Oracle 支持几乎 Oracle 数据库家族的所有平台之间的表空间移动。该功能具有如下一些优点：

- 在不同的内容提供者之间有效地发布数据。
- 在不同数据仓库、数据集市和 OLTP 系统之间轻松地移动数据。
- 轻松地跨平台迁移数据库。

注意：
该功能不完全支持当前的所有平台。可查看平台的相关文档以确定该平台是否符合条件。

接下来说明一些其他的问题。为了能够在平台之间移动表空间，必须将 compatible 设置为 10.0.0 或更高的值。设置该参数后，在下一次启动操作时，表空间数据文件将被设置为支持跨平台操作。

需要注意的是，只有在设置为可读写或联机时，只读文件和脱机数据文件才可支持跨平台操作。

14.5.2 字节排序和数据文件转换

本节将首先介绍 Oracle 数据库数据文件中字节排序的概念，以及这种概念如何影响在不同平台之间传送表空间。然后，我们将介绍如何根据需要转换这些数据文件。

1. 数据文件字节排序

Oracle 平台一般使用两种不同的字节排序方案(称为尾数格式)。如果多个平台使用相同的字节排序方案，则可以使用以前的常用方法传送表空间，而不会产生任何问题。

如果字节排序方案不同，则需要在 RMAN 中使用 convert 命令来将表空间转换为目标平台上所需的格式。可以通过连接动态视图 V$DATABASE 和新的 V$TRANSPORTABLE_PLATFORM 视图来确定尾数格式，如同下面的示例所示：

```
SQL> Select endian_format
  2  From v$transportable_platform tp, v$database d
  3  Where tp.platform_name=d.platform_name;

ENDIAN_FORMAT
--------------
Little
```

在该示例中，当前系统使用小尾数格式。因此，如果在两个系统上的查询返回相同的结果，则表示有兼容的数据文件格式；如果返回不同的结果，则需要使用 RMAN 和 compatible 参数来传送表空间。

2. 使用 RMAN 转换表空间尾数格式

如果需要转换另一个平台的表空间，则可以使用 RMAN。首先，创建保存转换文件副本的目录。在下面示例中，我们将使用目录路径 c:\oracle\oradata\betatwo。接下来，将希望转换的表空间设置为只读模式。然后，启动 RMAN 并使用新的 convert tablespace 命令，如同下面的示例所示：

```
Rman target=/
RMAN> convert tablespace users
to platform = ' AIX-Based Systems (64-bit)'
db_file_name_convert='c:\oracle\oradata\betatwo',
'c:\oracle\admin\transport_aix';
```

也可以转换目标站点上的数据文件：

```
Rman target=/
RMAN> convert datafile = c:\oracle\oradata\betatwo\*'
from platform = ' AIX-Based Systems (64-bit)'
db_file_name_convert='c:\oracle\oradata\betatwo',
'c:\oracle\admin\transport_aix';
```

使用的平台名来自于 V$TRANSPORTABLE_PLATFORM 视图的 PLATFORM_NAME

列。Oracle 对于放入正确的名称有严格的要求。

一旦完成了转换，就可以通过如下方式完成移动：使用可传送的表空间手动移动数据文件/表空间。需要注意的是，如果尾数格式不同，RMAN 就不能够帮助您完成操作。如果尾数格式相同，则可以阅读下一节以查看是否可以使用 RMAN 的新功能来跨平台移动数据库。

14.5.3　跨平台移动数据库

Oracle Database 10*g* 中的 RMAN 提供了全新的功能以帮助在尾数字节格式相同的平台之间移动数据库。convert database 命令结合 DBMS_TDP 包可以减少在平台之间移动数据库的整体工作负载。该进程包括如下步骤：

(1) 以只读方式打开数据库：

```
startup mount;
alter database open read only;
```

(2) 使用 dbms_tdb.check_db 进程来检查数据库状态。应该已经确定了平台的名称(通过 V$TRANSPORTABLE_PLATFORM 视图的 PLATFORM_NAME 列)。该程序应该在打开 serveroutput 命令时运行，如同下面的示例所示：

```
set serveroutput on
declare
      db_ready boolean;
begin
  db_ready := dbms_tdb.check_db
  ('Microsoft Windows IA (32-bit)',dbms_tdb.skip_readonly);
end;
/
```

(3) 使用 dbms_tdb.check_external 进程来标识外部对象：

```
set serveroutput on
Declare
     external boolean;
begin
     external := dbms_tdb.check_external;
end;
/
```

(4) 当数据库可以传送时，就可以使用 RMAN 的 convert database 命令。RMAN 创建数据库移动所需的脚本，但不真正执行移动操作，而是创建移动所需的文件：

```
CONVERT DATABASE NEW DATABASE 'copydb'
        transport script 'c:\oracle\copydb\copyscripts'
        to platform 'Microsoft Windows IA (32-bit)';
```

可选参数 db_file_name_convert 允许用户为需要转换的数据文件定义目录文件名。如同下面的示例所示。

```
CONVERT DATABASE NEW DATABASE 'copydb'
        transport script 'c:\oracle\copydb\copyscripts'
        to platform 'Microsoft Windows IA (32-bit)'
        db_file_name_convert
        'C:\ORACLE\PRODUCT\10.2.0\ORADATA\ROB10R2','C:\ORACLE\NEWDBDEST';
```

一旦完成该命令，只需遵循 RMAN 提供的指示就可以在目标数据库上完成转换的进程。

14.6 处理 RMAN 的异常情况

有时您就不能得到一个喘息的机会。在某些少有的情况下，RMAN 就是不正常工作。怎么办？下面是一些建议，如果 RMAN 还原/恢复不按所想要的操作进行，我们建议尝试如下建议。

如果还原操作不成功，则执行 crosscheck 命令，并确保 RMAN 认为可用的备份集确实是可用的。crosscheck 命令是用来确保 RMAN 目录和控制文件与磁盘上的实际内容是同步的。我们将在第 17 章中详细讨论 crosscheck 命令。在 restore 命令不工作，或是选择一个错误的备份集的情况下，运行下面所示的命令：

```
crosscheck backup of database;
crosscheck backup of archivelog all;
crosscheck archivelog all;
```

如果备份集显示的是 EXPIRED 状态，那么可能会丢失执行还原所需要的备份集。导致这种情况有很多原因—磁盘驱动器的丢失，或一个好心的系统管理员已经把备份集备份到磁带。

如果备份集没有问题，但还原操作仍不能正确运行，那么进行检测，例如，如果正在进行时间点还原，则检测设置的 UNTIL TIME 参数是否有问题。真实案例：在一个案例中，一位 DBA(顺便说一句，人很聪明)试图从 04/01/09 上获得的备份进行还原。但是，四个小时后他才意识到恢复字符串读取的是 03/01/09。像这样小的语法错误，就可以使人发疯。

通常情况下，错误消息可以提供一些关于问题的线索。例如，它们可以指出介质管理层的不正确配置。这是一个普遍问题。特别是将一个备份还原到数据库服务器而不是备份原来产生的地方。

如果 restore 命令运行成功，而 recover 命令运行失败，则再一次检测语法错误。这是另一个得到日期是错误的地方。并确保所需要的归档重做日志是可用的。我们不时看到的一个问题是，一些需要的重做结束了却没有归档。有时不得不去联机重做日志重新进行重做。像这样的情况，我们实际需要的是进入 SQL*Plus 并手动完成恢复进程。

使用 SQL*Plus 提示符手动执行恢复，执行如下操作：

(1) 执行普通的 RMAN restore 命令，最好使用 until time 子句。

(2) 执行普通的 RMAN recover 命令。

(3) 如果第(2)步失败，则从一个单独的终端窗口启动 SQL*Plus。

(4) 确定还原所需要的归档重做日志。可以使用 V$ARCHIVED_LOG 视图和 V$LOG 视图确定归档重做日志的位置，创建时间和相关联的日志序列号。下面是查询 V$ARCHIVED_LOG 视图，V$LOG 视图和 V$LOGFILE 视图的示例，这三个视图给出了完成恢复所需的一些重要信息：

```
-- This tells us the last archived redo log file sequence number.
select max(sequence#), max(resetlogs_time), max(resetlogs_change#)
from v$log_history
where resetlogs_time=(select max(resetlogs_time) from v$log_history);

-- This tells us the online redo log sequence numbers.
select a.sequence#, a.first_time, b.member
from v$log a, v$logfile b
where a.group#=b.group#
order by a.sequence#;
```

(5) 命令 recover database 指示希望恢复的归档重做日志。在某些情况下，可能需要输入联机重做日志的名称来恢复数据库，因为其中包含的重做却没有归档。在这种情况下，我们从联机重做日志的 C:\ORACLE\ORADATA\BETA1\REDO02.LOG 目录中应用重做。

```
SQL> recover database
ORA-00279: change 5071334 generated at 08/17/2008 15:35:51 needed for
thread 1
ORA-00289: suggestion :
/oracle01/flash_recovery_area/ORCL/archivelog/2008_08_17/
o1_mf_1_5_4bk6onh8_.arcORA-00280:
change 5071334 for thread 1 is in sequence #5
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
C:\ORACLE\ORADATA\BETA1\REDO02.LOG
Log applied.
Media recovery complete.
```

(6) 然后打开数据库：

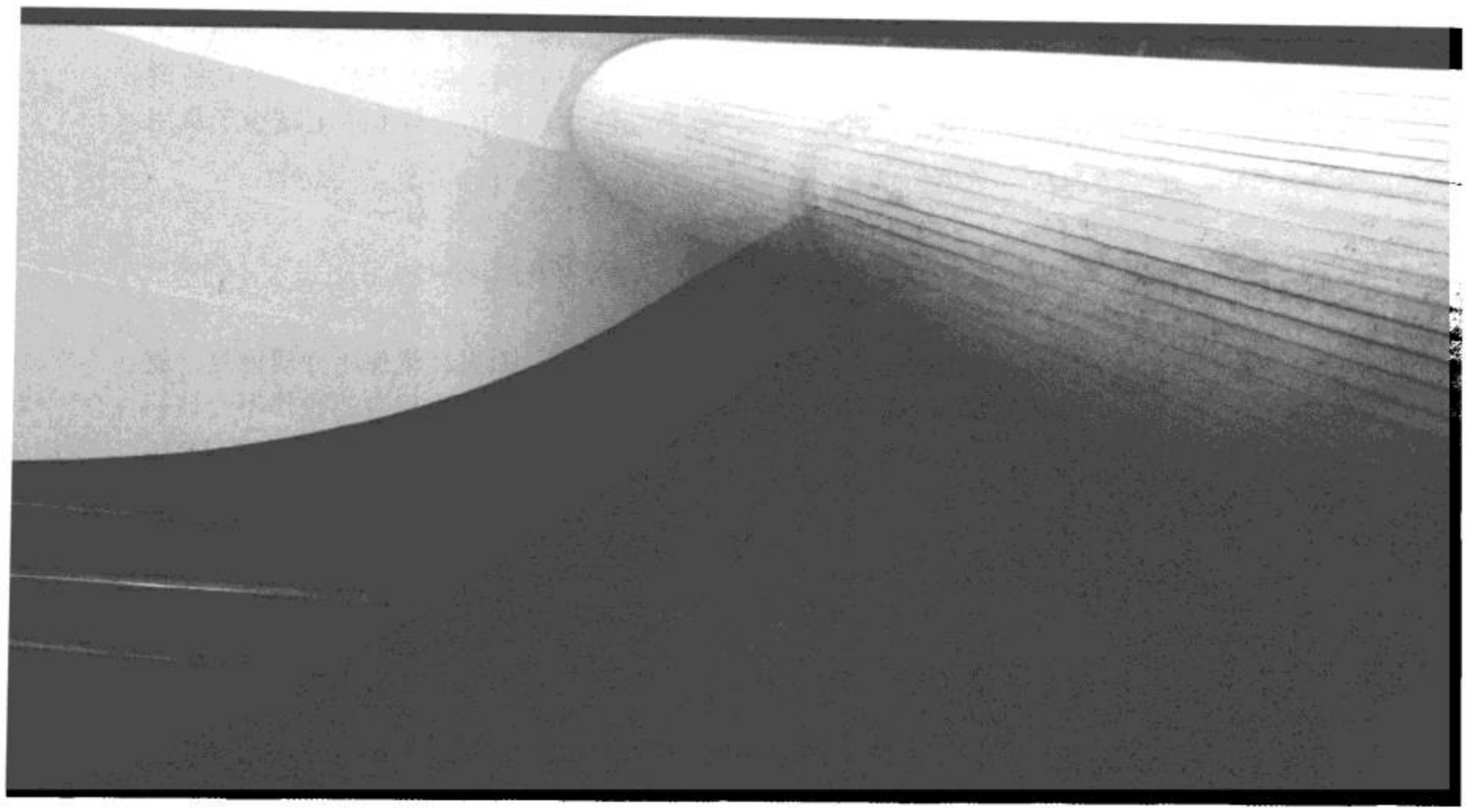
```
SQL> alter database open;
Database altered;
```

14.7　小结

本章介绍了 RMAN 中可用的时间点恢复。RMAN 支持基于时间、SCN，以及基于删除的恢复。本章也介绍了 RMAN 通过使用 resetlogs 命令执行恢复操作的功能，对于过去一直被相关问题困扰的 DBA 来说，这是非常有用的一项功能。

本章也介绍了各种恢复主题，包括归档重做日志的恢复、只读表空间恢复的注意事项，以及块级别的恢复。简而言之，我们广泛介绍了 RMAN 的各种恢复工具。

在本章的最后，我们介绍了表空间时间点恢复。对于表空间时间点恢复，可能会由于各种原因而增加操作的复杂性，因此最好参考 Oracle 文档以了解关于实际情况的特定实现。



第15章

从用户错误中恢复：闪回技术

RMAN中的介质恢复提供了针对所有无法预料的问题的关键保护措施——数据块损坏、硬件故障、甚至是完全的数据库丢失。但是，本书到目前为止尚未介绍使用介质恢复操作的最大原因：用户错误。

用户错误可概括地定义为由人为过失(而不是软件或硬件故障)造成的错误，例如使用错误的值更新表、删除表、或者截断表。这些错误远比硬件故障常见(虽然人为错误始终被称为硬件错误)。一般来说，用户错误分类为逻辑错误——这种错误是数据本身存在的逻辑错误，并且使用介质恢复选项对此执行的修正通常较为昂贵。

本章中将讨论 Oracle Database 11g 中针对逻辑错误的编程准备以及从逻辑错误中恢复的新方法。这些方法包括一些 Oracle 10g Database 中已有的功能，但是对这些功能进行了扩展(并且更加易于使用)；也包括一些全新的功能，这些功能从根本上减少了从用户导致的灾难中恢复所花费的时间。

15.1 为不可避免的问题做准备：闪回技术

当遇到逻辑错误时，不应该立刻就考虑使用介质恢复。如果经常使用介质恢复，就会导致大规模的停机。一般来说，用户错误是不可以恢复的，因为相应的操作不会被解释为数据库错误。例如，“Delete * from scott.emp”就不是错误，它是完全合理的 DML 语句，该语句被适当地记录在重做流中。因此，如果还原数据文件，然后执行恢复，则只需要再次执行 delete * from scott.emp。时间点恢复可能是一种解决方案，但对于想以此来避免完全还原数据库的 DBA 来说不适合——这种方法会产生太多的停机。针对用户错误，表空间时间点恢复(TSPITR)提供了对用户错误执行介质恢复的缓和版本(较少的停机)，但它仍然需要一次表空间的完全停机，并需要大量用于保存临时实例副本的空间和对象级的限制(如高级队列表)。

为了帮助进行用户错误恢复并且改进 RMAN 的介质恢复，Oracle 在 Oracle Database 10g 中引入了闪回技术的概念。闪回技术涉及一组功能，这些功能提供了大量方法来从用户错误中恢复，并且使恢复过程简单而快速。这些闪回功能如下：

- 闪回查询(Flashback Query)
- 闪回表(Flashback Table)
- 闪回事务(Flashback Transaction——11g 中的新功能)
- 闪回删除(Flashback Drop)
- 闪回数据库(Flashback Database)
- 闪回数据归档(Flashback Data Archive——11g 中的新功能)

15.2 闪回查询

实际上 Oracle 9i 中也有一些闪回查询功能。在 10.2 版本中，对这些闪回查询功能进行了扩展和简化，从而可以更好地使用它们。更好地使用是指不依赖于 PL/SQL 接口，现在这些功能都内置在 SQL(有时是 RMAN！)中，从而不需要编写 PL/SQL 代码块来查看行的历史版本。

15.2.1 闪回和撤消段简介

前两种闪回类型(闪回查询和闪回表)的功能完全基于 Oracle Database 中长期存在的技术：撤消段(以前称为回滚的段)。撤消段用于撤消没有提交的事务。在过去，已经提交的事务不可以撤消，因为回滚段中与行关联的“前”映像已经被释放，用于将来的重

写，因此前映像随后不一定能找到。

当提交事务时，撤消段中保存行的前映像的盘区被释放，用于将来的重写。不同于 Oracle 9*i* 中回滚空间的使用方式，它在重写前面使用的段之前，所有新的事务都将查找撤消表空间中未使用的空间。尽管如此，事务总是会使用存在时间最长的剩余盘区。这意味着数据库中行的前映像将比过去存在更长的时间——从而我们可以在过去的事物中找到撤消段。

好消息是，在 Oracle 9*i* 和以后的版本中，Oracle 开始使用闪回查询技术。我们现在可以控制撤消盘区在被重写之前所保留的时间。完成这项工作后，我们就可以很好地使用撤消——从而帮助我们撤消已提交的错误事务。

根据撤消盘区在被重写之前保留在撤消表空间中的时间来决定是否可以查询对象或将对象改回到过去的某个时间点。新的事务基于撤消表空间中的空间压力来使用撤消盘区。一般来说，Oracle 只有在用完所有其他可能的盘区后才会重写撤消盘区——也就是在利用撤消表空间中的每个盘区后才会进行重写操作。然后，Oracle 查找存在时间最长的盘区并重写它。至于在回退到多久以前的情况下可以使用“闪回查询/闪回表”，则取决于 Oracle 如何从提交事务的时间点切换到重写该事务的撤消时间点。从提交事务到重写撤消盘区的周期就是闪回窗口(flashback window)。

有许多因素决定闪回窗口，但最重要的因素是事务负载。可以使用视图 V$UNDOSTAT 查看撤消使用情况的统计。该视图中的每一行代表 10 分钟内利用的撤消块数量。通过峰值使用情况运行对该视图的一些分析，可以提供一个适当的模板，该模板可以指导您对撤消进行设置。

设置闪回查询和闪回表的撤消参数

使用闪回查询的指导原则要求首先启用自动撤消——不允许设置任何回滚段(实际情况并不是如此，也可以同时使用闪回操作和回滚段，但 Oracle 不鼓励这样做。所以不要再想尝试手动设置回滚段)。Oracle 通过使用强调保存事务历史信息的新算法来控制撤消管理，该算法不存在于回滚段中。因此，需要在参数文件(PFILE)或服务器参数文件(SPFILE)中设置 UNDO_MANAGEMENT=AUTO。其次，设置 UNDO_TABLESPACE 参数以指出哪个表空间将处理撤消任务。最后，将 UNDO_RETENTION 设置为以秒为单位的值，该值设置保留撤消段的所需时间长度。

15.2.2　执行闪回查询

执行表的闪回查询非常简单，现在已经将该操作集成到 SQL 中。我们只需要知道查看表内容的过去的时间点，然后将其插入到查询中：

```
select scr_id, head_config from ws_app.woodscrew as of timestamp
    to_timestamp('2009-06-27 04:27:00','YYYY-MM-DD HH:MI:SS')
    where scr_id=1001;

    SCR_ID HEAD_CONFIG
---------- --------------------
```

```
      1001 Phillips
      1001 Phillips
```

如果知道正在查找的更改的系统更改号(SCN)，也可以使用 SCN 限定符：

```
select scr_id, head_config from ws_app.woodscrew
 as of scn 751652 where scr_id=1001;

    SCR_ID HEAD_CONFIG
---------- --------------------
      1001 Slot
      1001 Slot
```

15.2.3 使用 Oracle 企业管理器执行闪回版本查询

使用 Oracle 企业管理器(OEM)可以更容易地实现闪回查询——以及它的相关技术，即闪回事务查询(Flashback Transaction Query)和闪回版本查询(Flashback Version Query)。使用 OEM 可以将闪回查询快速变为一个撤消用户错误的操作，无论是通过闪回表还是将 SQL 撤消语句应用于存在错误的事务。

OEM 结合了多种技术的最佳功能以提供一个用户界面，该用户界面可帮助用户快速获得解决方法。该用户界面在后台使用事务查询来进行较为完全的调查以发现逻辑错误。在多种技术中，第一个技术要介绍的就是闪回版本查询，也称为行历史(row history)。闪回版本查询能够查找指定时间框架内某一行的每个版本。因此，我们提供查找行的查询，并且提供希望检查的时间框架，Oracle 就会返回该行经历的每次迭代的列表。通过该列表可以了解这段时间以来行的变更，从而确定问题的根源所在。

RMAN 实习：研究闪回版本查询

实习注意事项

该实习要求构建一些表并且使用一些行填充这些表，从而可以观察闪回版本查询的实际操作。下面是 WOODSCREW 表及其索引的 DDL 和 DML。这段代码也构建了一个辅助表，其中的一些行将在后面用于闪回删除(Flashback Drop)和闪回数据库(Flashback Database)。不必非得使用这里的简单小测试，可以使用系统中已有的表来简单地进行测试。

```
create table woodscrew (
scr_id            number not null,
manufactr_id       varchar2(20) not null,
scr_type          varchar2(20),
thread_cnt         number,
length            number,
head_config        varchar2(20));
alter table woodscrew add primary key
  (scr_id, manufactr_id) using index;
```

```
create index woodscrew_identity on woodscrew
(scr_type, thread_cnt, length, head_config);

create table woodscrew_inventory (
  scr_id          number not null,
  manufactr_id     varchar2(20) not null,
  warehouse_id     number not null,
  locale          varchar2(20),
  count           number,
  lot_price    number);

insert into woodscrew values
(1000, 'Tommy Hardware', 'Finish', 30, 1.5, 'Phillips');
insert into woodscrew values
(1000, 'Balaji Parts, Inc.', 'Finish', 30, 1.5, 'Phillips');
insert into woodscrew values
(1001, 'Tommy Hardware', 'Finish', 30, 1, 'Phillips');
insert into woodscrew values
(1001, 'Balaji Parts, Inc.', 'Finish', 30, 1, 'Phillips');
insert into woodscrew values
(1002, 'Tommy Hardware', 'Finish', 20, 1.5, 'Phillips');
insert into woodscrew values
(1002, 'Balaji Parts, Inc.', 'Finish', 20, 1.5, 'Phillips');
insert into woodscrew values
(1003, 'Tommy Hardware', 'Finish', 20, 1, 'Phillips');
insert into woodscrew values
(1003, 'Balaji Parts, Inc.', 'Finish', 20, 1, 'Phillips');

insert into woodscrew_inventory values  (
(1000, 'Tommy Hardware', 200, 'NORTHEAST', 3000000, .01);
insert into woodscrew_inventory values  (
(1000, 'Tommy Hardware', 350, 'SOUTHWEST', 1000000, .01);
insert into woodscrew_inventory values  (
(1000, 'Balaji Parts, Inc.', 450, 'NORTHWEST', 1500000, .015);
insert into woodscrew_inventory values  (
(1005, 'Balaji Parts, Inc.', 450, 'NORTHWEST', 1700000, .017);
commit;
```

步骤 1　打开 OEM 控制台的数据库主页，并且选择 Schema | Tables。这会打开用户模式中所有表的视图，以该用户的身份登录系统。通过将 Schema 改为用户并且单击 Go 按钮，可以将其改为 WOODSCREW 表的属主。然后，可以选择 WOODSCREW 表并从 Actions 下拉列表框中选择 View Data。单击 View Data 后，OEM 将显示如图 15-1 所示的视图。需要注意的是，HEAD_CONFIG 列中的值都是 Phillips。

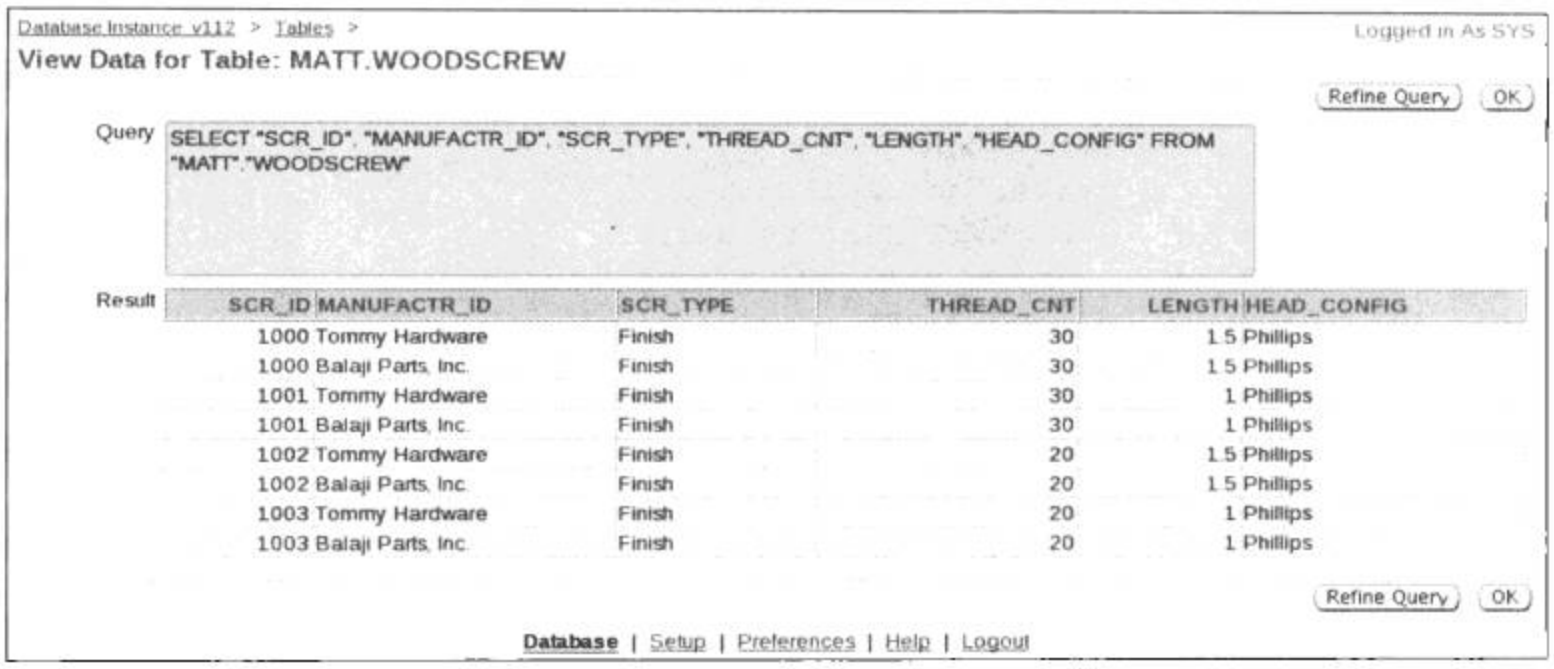


图 15-1 视图

步骤 2 更改表中的行以反映木螺钉(woodscrew)的不同头配置。

```
update woodscrew set head_config= 'Slot'
where scr_id=1001;
commit;
```

步骤 3 查看表中的新数据。在 Tables 视图中，选择 WOODSCREW 表，从 Actions 下拉列表框中选择 View Data，然后单击 Go 按钮。在图 15-2 中需要注意的是，其中两行的 HEAD_CONFIG 值是 Slot 而不是 Phillips。

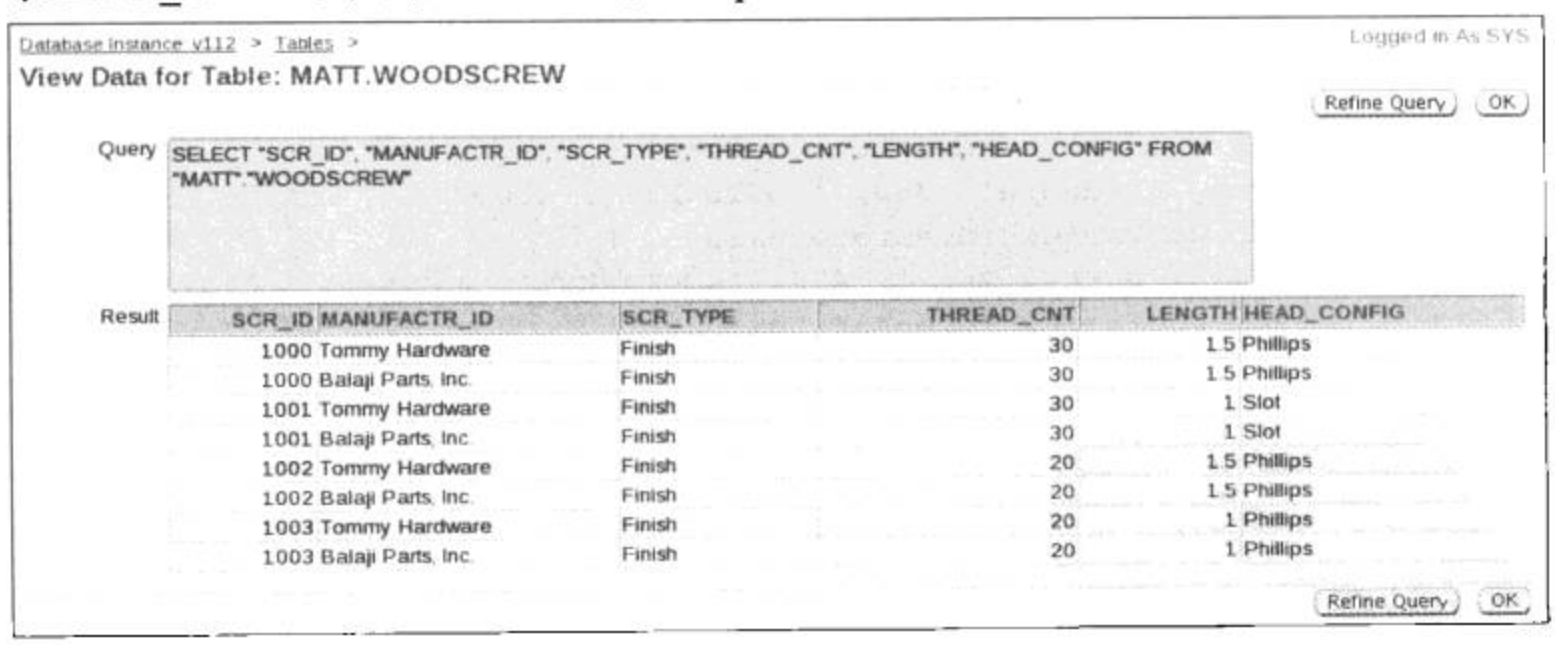


图 15-2 查看 WOODSCREW 表的数据

步骤 4 在木螺钉企业组织中，可以确定 scr_id 为 1001 的螺钉是槽头(slot-headed)螺钉，而不是十字头(Phillips)螺钉。此时在数据库中引入了一个逻辑损坏。接下来检查一行并查看该行经历的各种版本。在 Tables 视图中，选择 WOODSCREW 表，从 Actions 列表中选择 Flashback Versions Query，然后单击 Go 按钮，打开 Perform Object Level

Recovery 向导。

步骤 5　在该向导中，需要提供闪回查询的参数。首先，通过选择该向导的步骤 1 页面中的 Move All 选项，选择所有的列。单击该页面右边的 Next 按钮。在步骤 2 页面中，指定隔离某一行的子句。我们将使用如下的 WHERE 子句：

```
where scr_id=1001 and manufactr_id='Tommy Hardware'
```

指定子句并单击 Next 按钮后，可看到行的不同版本，以及选择 SCN 的不同选项以恢复到的某个时间(时间由我们决定)。图 15-3 所示的屏幕截图显示了查询结果，其中有更新两行的事务。单击 Transaction ID 可看到具体的事务(后面将介绍闪回事务(Flashback Transaction)功能)。

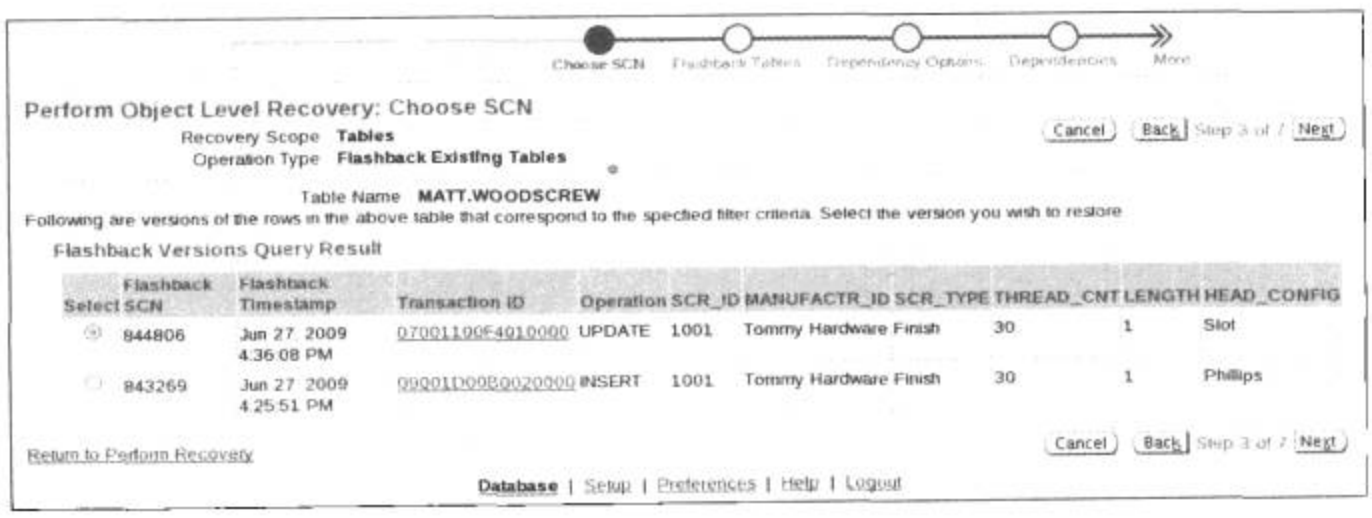


图 15-3　查询结果

步骤 6　检查对数据库中的行执行的不同操作，确定哪一行可能存在错误。从图 15-3 中可以继续向导并执行表的闪回。

步骤 7　如果希望证明 OEM 正在执行操作，可单击 Back 进入 Recovery:Row History Filter 页面。在该页面的底部有一个 Show Flashback Versions Query SQL 按钮；单击该按钮后，可以看到大量的 SQL 代码，如图 15-4 所示。

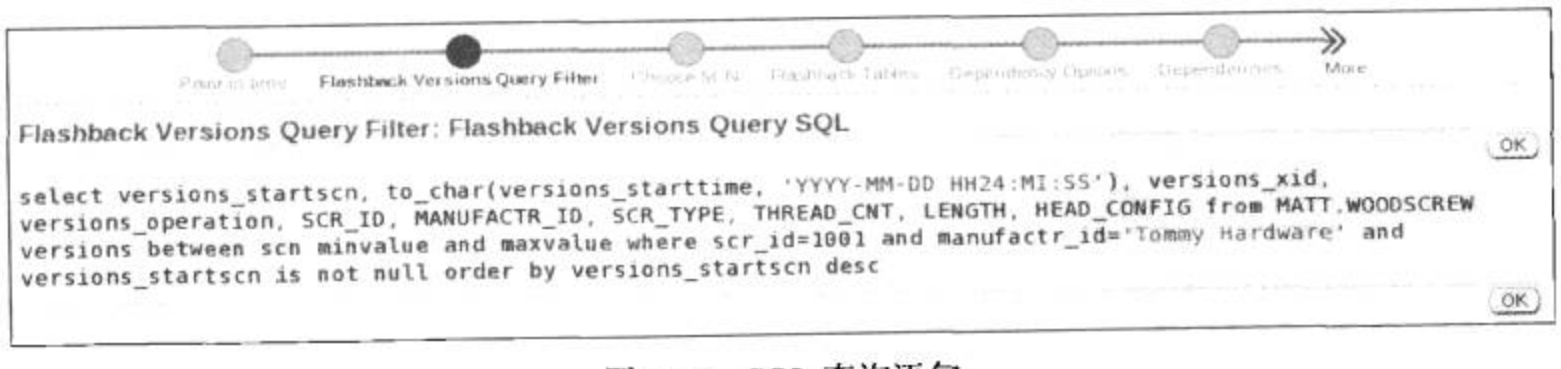


图 15-4　SQL 查询语句

15.3　闪回表

或许最引人注目的闪回技术是以简单而直观的方式将表恢复到以前的时间点。在单

个表或一组表上执行时间点恢复的能力是构建整个数据库副本的基础——从而可以析取单个表，然后重新导入到产品数据库中。使用闪回表(Flashback Table)可以消除不必要的克隆操作。

闪回表使用与闪回查询相同的机制——使用存储在撤消段中的信息。Oracle 可以按照一次一个事务的方式回绕数据库，从而将表恢复到过去的指定时间点。闪回表操作取决于撤消，因此具有相同的限制：只可以将表闪回到撤消段所允许的过去的时间点。

除了撤消之外，闪回表还需要启用表的行移动(row movement)。行移动最初是提供分区表的功能，如果更新操作更改了分区键值，使用行移动功能就可以将更新的行移动到适当的分区。闪回表使用行移动功能来辅助回绕操作。可以使用如下的 alter table 命令启用行移动功能。

```
alter table woodscrew enable row movement;
```

闪回表并不对所有的用户错误都有效。针对表执行的某些 DDL 操作不可以被撤消。最为重要的是，不可以将表闪回到执行 truncate table 操作之前，因为 truncate 不会产生任何撤消——这就是 truncate 存在的原因，它与 delete * from table 相对。同样，闪回表不可以用于已经删除的表(此时使用闪回删除，15.5 节中将对此进行介绍)。

15.3.1 从 SQL 中执行闪回表操作

启用行移动功能后，我们就可以继续对表执行常规的操作。然后，当表中发生用户导致的损坏时，就可以在命令行中使用 SQL 来执行闪回表操作：

```
flashback table matt.woodscrew to timestamp
    to_timestamp('2009-06-29 13:30:00','YYYY-MM-DD HH24:MI:SS')
```

另外，如果能够确定 SCN(例如，通过闪回查询得到)，则可以使用该 SCN：

```
flashback table matt.woodscrew to scn 751652;
```

类似于闪回查询，闪回表操作的性能取决于已经回绕的数据数量，以及回绕的过去的时间点。撤消的数据越多，该操作花费的时间就越多。但这总是比通过其他方法执行表的时间点恢复要快。例如，尝试 TSPITR，或者尝试将表空间还原到不同的实例，然后从克隆实例中导出表并导入到产品数据库中。没有比闪回表性能更佳的方法。

15.3.2 在 Oracle 企业管理器中使用闪回表

OEM 内部的功能可利用闪回表特性，该附加功能首先通过闪回版本查询(Flashback Versions Query)对表进行研究，从而确定希望闪回的确切时间。如果已经知道闪回的确切时间，则在命令行中使用 SQL 就和在 OEM 中使用闪回表向导一样简单。然而，OEM 确实提供了一种方法来确定哪种依赖起作用，如下面的 RMAN 实习所示。

启用行移动和闪回表

预见可能的闪回表候选物并且尽可能启用行移动，这一点至关重要。不可以先启动

行移动，然后将表闪回到启用行移动之前的时间点。这种操作将导致如下错误：

```
ORA-08189: cannot flashback the table because row movement is not enabled.
```

换句话说，不可以等待到需要闪回表时，才启用行移动作为闪回操作的一部分。我们可能需要使用闪回事务查询来手动撤消每个行更改。

RMAN 实习：研究闪回表

实习注意事项

在该实习中，我们将“无意中”删除 WOODSCREW 表中的所有行，然后将整个表闪回到删除事务发生之前的时间点。

步骤 1　查看 WOODSCREW 中的数据。由于前面的练习，我们有必要截断该表，然后使用在 15.2.3 小节的 RMAN 实习“研究闪回版本查询”中介绍的脚本手动地重新插入记录。应确保具有所有 8 行，并确保在插入错误之前启用了行移动：

```
alter table woodscrew enable row movement;
```

步骤 2　插入错误。我们将使用 SQL *Plus 的 DELETE 语句删除表中的所有行。在此之后，从 OEM 中的 Actions 下拉列表框中选择 View Data，查看空表。

```
delete from woodscrew;
commit;
```

步骤 3　在 OEM 数据库主页中，选择 Schema| Tables，然后选择 WOODSCREW 表。在 Actions 下拉列表框中，选择 Flashback Table，然后单击 Go 按钮。这会打开 Perform Object Level Recovery:Point-in-Time 向导，该向导首先要求您指定恢复的时间点。我们假设不知道删除操作何时发生，因此选择第一个选项以估计行更改和事务，单击 Next 按钮。

步骤 4　现在可以看到在本章的 RMAN 实习中常见的屏幕——闪回版本查询界面。在该界面中，我们必须设置列(MOVE ALL)和 WHERE 子句，从而缩小搜索范围。因为删除操作影响了所有行，在这里我们可以选择一行进行检查。我们将使用与前一个实习相同的行：

```
where scr_id=1001 and manufactr_id='Tommy Hardware'
```

现在我们可以看到有关 DELETE 操作的信息，该 DELETE 操作删除了 WOODSCREW 表中的所有行，如图 15-5 所示。

步骤 5　我们可以单击 Transaction ID 来检查整个删除操作。然而，此处更为重要的内容是 Flashback SCN 列，该列显示了设置用于撤消 DELETE 操作的 SCN。从该列表中选择适当的 DELETE 事务，单击 Next 按钮就可以自动选择该屏幕上指定的闪回 SCN。

步骤 6　在闪回表向导的步骤 4～步骤 7 中，Oracle 允许我们指定任何逻辑相关的对象，这些对象应该回绕到与表相同的 SCN。Oracle 自动实施表中已有的所有约束，但是我们可能有一些应该闪回的逻辑相关表。如果是这样，则在此处就可以指定这些表。在该示例中，我们没有任何相关的表，因此单击 Next 按钮。

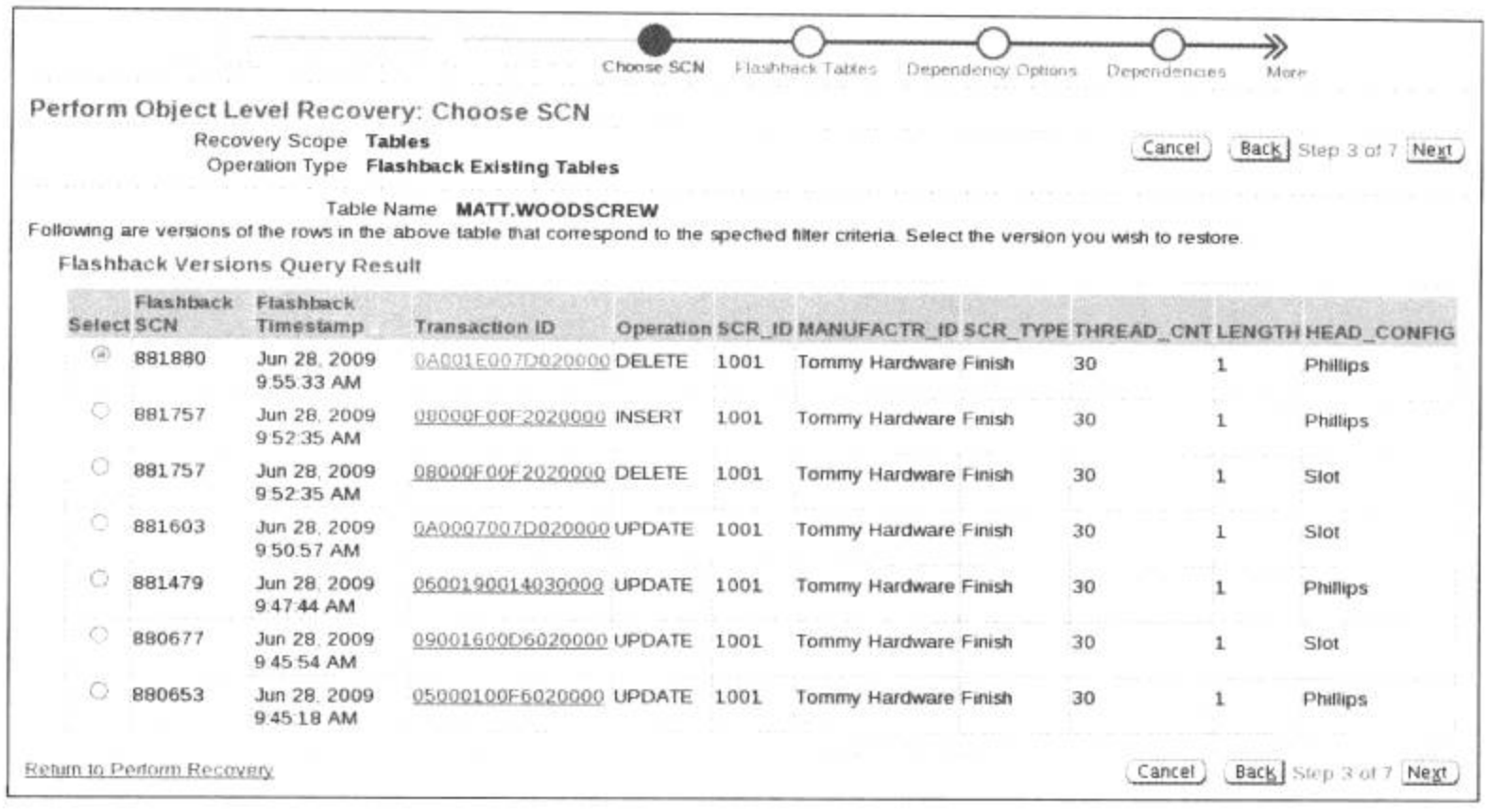


图 15-5 查询结果

步骤 7 此时会直接显示该向导的 Step 7 页面，在其中可以看到将执行的具体操作的概述。在此处最有用的是 Show Row Changes 和 Show SQL 按钮，单击这两个按钮可分别显示更改了哪些行以及 OEM 将自动执行哪些 SQL 语句。单击 Submit 按钮。如果存在任何问题，或者只是觉得这样做更好，则可以将 OEM 生成的 SQL 剪切到 SQL *Plus 会话中以运行闪回表操作：

```
FLASHBACK TABLE MATT.WOODSCREW TO SCN 804109;
```

15.4 闪回事务

寻找数据库中坏数据的方法往往不止一种。使用闪回事务可以看见某一指定事务产生的所有更改或者某个特定时间框架内所有的事务。然后只需撤消一个事务级错误，而不用回滚整个表。使用闪回事务可以保存由于操作错误导致的所有其他更改。这些操作包括消除最低可能性的错误和保留良好的数据。也可以闪回一个事务子集，而不必撤消整个事务。

不同于闪回版本查询(Flashback Versions Query)，闪回事务(Flashback Transaction)不是使用撤消段来理解收回一个错误所需进行的操作，而是使用重做以及利用日志挖掘(LogMiner)功能挖掘出事务的更改向量，然后确定回滚这些更改的最佳方式。

使用闪回事务的最好方式就是使用 OEM，可以在 OEM 控制台为数据库保存灾难和日志，并且使用在 RMAN 实习“研究闪回版本查询”中描述的同样操作来回退到 WOODSCREW 表中，此时可看到该表中一个可用的操作就是闪回事务(Flashback Transaction)

操作。

在使用闪回事务前，需要为所查询的表打开补充日志数据(supplemental log data)。与启用闪回表的行移动一样，如果在需要使用闪回事务之前而没有打开补充日志数据，那将为时已晚了。撤消段不包含执行闪回所需的信息。如果使用闪回事务，在 OEM 中将看到一个错误消息，该错误会告知您需要执行的操作，如图 15-6 所示。

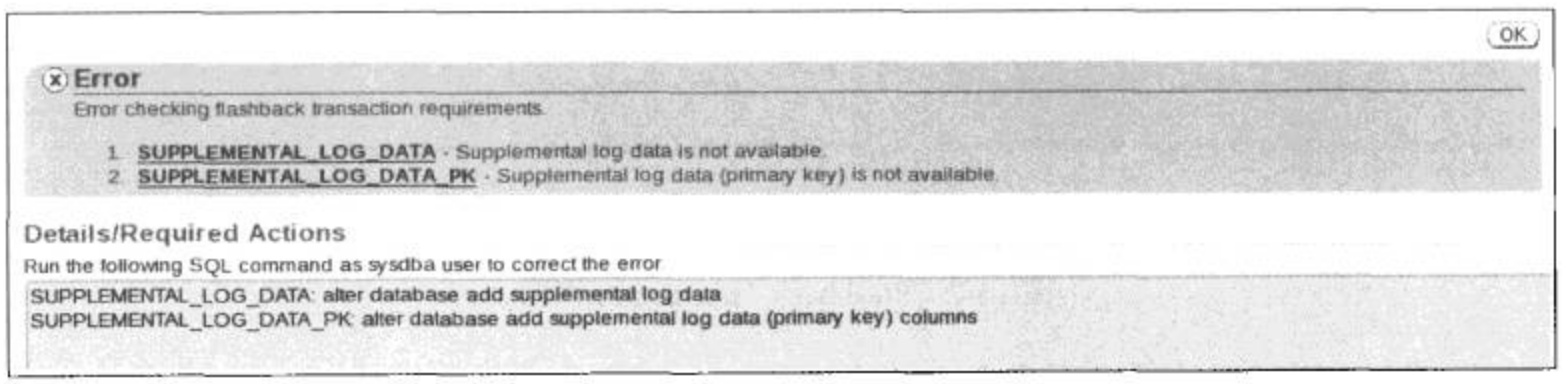


图 15-6　错误提示

闪回事务查询非常引人注目，因为即使窗口中可能仅仅只有一行错误，使用它可全面地检查出一个坏事务。例如，如果发现已删除 WOODSCREW 表中的一行，则在闪回版本查询中能查看到那行。然后，可以获得删除操作的 Transaction ID，同时可看到删除了多少行。该查询提供了在全部范围内的查看。

RMAN 实习：在企业管理器中使用闪回事务

实习注意事项

在该实习中，我们引入了一个错误操作，然后使用闪回事务撤消该操作。使用的表是本章前面实习中使用的 WOODSCREW 表。

步骤 1　启用 WOODSCREW 表中的补充日志数据：

```
SQL> alter database add supplemental log data;
SQL>atter database add supplemental log tata (primary key) columns;
```

步骤 2　在 WOODSCREW 表中引入一个错误操作：

```
SQL>delete from woodscrew where scr_id='1001';
commit;
```

步骤 3　当查询由 Tommy 制造的一英寸的木螺钉时，看到螺钉已经从数据库中删除，即使公司仍然向用户提供这些螺钉，但实际上发生了什么？在 OEM 中，选择 OEM Database Home | Schema | Tables 页面，选择 WOODSCREW 表属主，然后单击 Go 按钮。

步骤 4　选择 WOODSCREW 表，从 Action 列表框中选择 Flashback Transaction 选项，然后单击 Go 按钮。从 Query Time Range 中选择一个时间范围，其匹配可能的删除时间。然后单击 Go 按钮，并且当 OEM 短暂挖掘日志文件时等待查询表中所有事务的信息。然后，会显示可用的闪回事务，如图 15-7 所示。

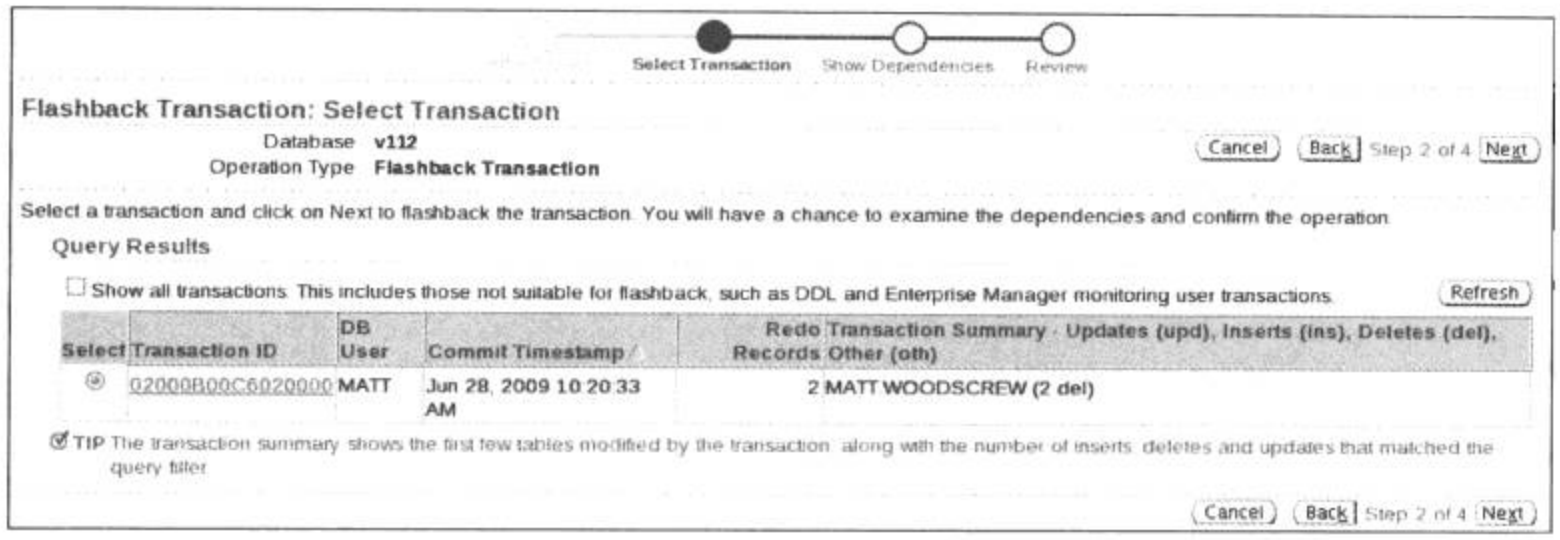


图 15-7 Flashback Transaction 视图

步骤 5 从可用的闪回事务表中单击已链接的 Transaction ID，此时可以看到与该事务关联的每个更改向量的细节。这可以提供指定的 SQL 语句，该语句用于执行每个不符合要求的操作，同时还可为撤消整个事务中一个特定的子集提供一个路线图。单击 OK 按钮，回退到闪回事务(Flashback Transaction)向导，如图 15-8 所示。

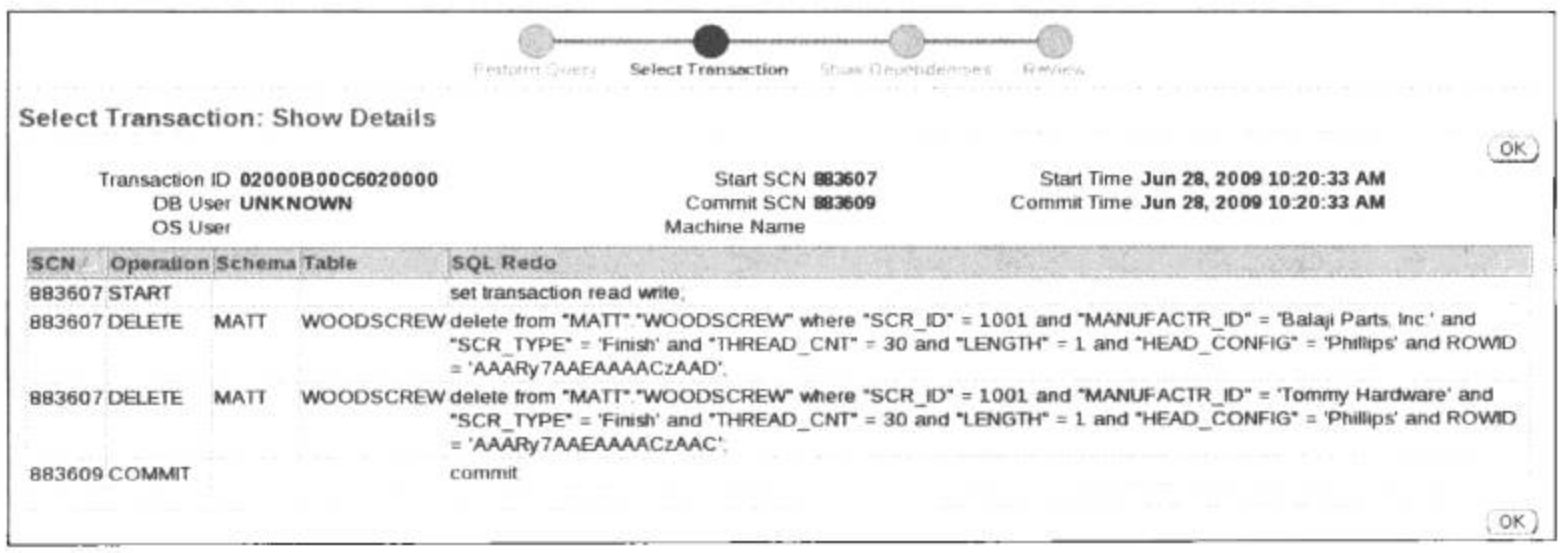


图 15-8 Select Transaction 视图

步骤 6 单击 Next 按钮，闪回事务将最终完成操作，该操作需要求完成回滚。这可能会花几分钟完成操作。当完成操作后，可以获得一个最终的操作检查页以及一个可运行任何后续 SQL 的复选框，该 SQL 要求在提交事务之前确认更改。然后，单击 Finish 按钮。

步骤 7 OEM 确认已选择的事务已经完全的回退。单击 OK 按钮，然后可回到 Tables 页面，在该页面可在此看到 WOODSCREW 中的数据，确认删除操作已经回退。

15.5 闪回删除

闪回删除(Flashback Drop)允许我们“撤消删除”数据库对象。不会再出现迫切希望

将整个数据库还原到过去的时间点的用户，因为这些用户认为他们使用的是 DEV 实例，而不是 PROD 实例。

对于实现闪回删除的过程，没有需要特别介绍的内容。在 Oracle Database 10g 中，当删除表时，仅仅是将其重命名为系统可以识别的字符串，但是相应的段仍应保留在删除该表的表空间中，直到撤消删除对象或手动清除该对象时，或者表空间中常规对象的空间用完时才会删除该段。如果表空间中的空间不足，Oracle 将根据存在时间的长短来删除存在时间最长的对象。

当删除对象时，Oracle 不只是重命名该对象。所有相关的对象也会移动到回收站(Recycle Bin)中：索引、触发器以及约束。因此，当撤消删除表时，也会撤消对所有相关对象的操作。

15.5.1　回收站

回收站(Recycle Bin)是数据库中所有已删除对象的虚拟目录——即已经删除但还没有清除的对象列表。回收站是一个逻辑容器，它不需要特定的存储位置；所有已删除对象的实际存储位置仍是对象被删除之前所在的表空间。例如，用户 matt 删除了表 WS_APP.WOODSCREWS。WOODSCREWS 表位于表空间 WS_APP_DATA 中，但它的两个索引位于 WS_APP_IDX 表空间中。当删除 WOODSCREWS 时，将该表重命名为内部的名称，对该表上已有的两个索引也执行相同的操作。这些操作都显示在 DBA_RECYCLEBIN 视图中。然而，实际的 WOODSCREWS 表段仍然存在于 WS_APP_DATA 表空间中，而相应的两个索引也存在于 WS_APP_IDX 表空间中。它们都是回收站的逻辑部分，但物理存在于它们一直所在的位置。

可以通过如下两个数据字典视图快速查看回收站：

- USER_RECYCLEBIN
- DBA_RECYCLEBIN

1. 清除回收站

不需要从回收站中手动地清除已删除的对象。在表空间中的其他段需要空间时，对象才从回收站中清除。换句话说，已删除的对象将继续占用表空间中的空间，直到表空间中的其他对象的空闲空间用完为止。然后，最先清除第一个删除的对象。Oracle 自动在清除表之前先清除索引，从而最后清除实际的数据。如果启用自动扩展，回收站对象就会在表空间自动扩展之前被删除。

可以使用新的 purge 命令清除回收站。可以清除用户、对象或表空间，或者清除整个回收站：

```
purge table matt.woodscrews;
purge index matt.woodscrews_pk_idx;
purge tablespace sales;
purge recyclebin;
```

对象在回收站中存放多久?

这是一个非常好的问题，其答案是：根据情况而定。也许真正问题是："我是否可以控制对象存在于回收站中的时间？"但对于这个问题并没有确切的答案。

如果删除对象的表空间中的空间不足，则不可以将已删除的对象强制保留在回收站中。即使启用自动扩展，也要在表空间扩展之前清除已删除的对象。因此，如果希望确定回收站中对象的确切生命周期，则有两个选项：建立过大的表空间以适应删除操作；或者手动管理回收站并清除不需要保存的对象，从而为需要保存的对象留出空间。

因此，我们可以缩短对象在回收站中停留的时间。但是，如果表空间存在空间不足的问题，则不可以强制将对象保留在回收站中。

2. 撤消删除回收站中的对象

从回收站中恢复对象非常简单——使用一条简单的 SQL 语句，将该对象重命名为原始名称，并且对任何相关的对象执行相同的操作：

```
flashback table ws_app.woodscrews to before drop;
```

当然，恢复对象有时并不简单。例如，如果有多个名称相同的已删除对象，则需要根据新的和改进的回收站名称引用对象：

```
SQL> select object_name, original_name, droptime,
    dropscn from user_recyclebin;
    OBJECT_NAME                    ORIGINAL_NAME
    ------------------------------ --------------
    DROPTIME            DROPSCN
    ------------------- ----------
    RB$$48623$INDEX$0              PK_WOODSCREW
    2004-01-12:15:21:26  1241651
    RB$$48622$TABLE$0              WOODSCREW
    2004-01-12:15:21:26  1241652
    SQL> flashback table " RB$$48622$TABLE$0" to before drop;
```

注意，回收站对象名称周围的引号。由于名称中有特殊符号，因此必须使用这些引号。

如果您已经删除了一个对象，然后创建具有相同名称的新对象，则仍然可以闪回第一个对象。当从回收站中恢复对象时，可以使用闪回 SQL 中的相关语法重命名该对象：

```
flashback table ws_app.woodscrews to before drop rename to woodscrews_history;
```

RMAN 实习：研究闪回删除和回收站

实习注意事项

在该实习中将删除 WOODSCREW 表并检查回收站内容。虽然可以在命令行中完成这些操作，但该实习将介绍如何使用 OEM 撤消删除表。

步骤 1 删除 WOODSCREW 表。在 OEM 中，选择 Schema | Tables，将模式改为

WOODSCREW 表的属主。从表的列表中选择 WOODSCREW 表，然后单击 Delete with Options 按钮。此时显示提供多个删除选项——删除表、删除行或者截断的屏幕。选择第一个选项(DROP)并单击 Yes 按钮。

步骤 2　查看表的列表。需要注意的是，WOODSCREW 不在该列表中。单击该屏幕右下角的 Recycle Bin 链接以访问回收站，如图 15-9 所示。首先，在 Search 部分中，输入 WOODSCREW 表属主的模式名(Schema Name)，并且单击 Go 按钮。

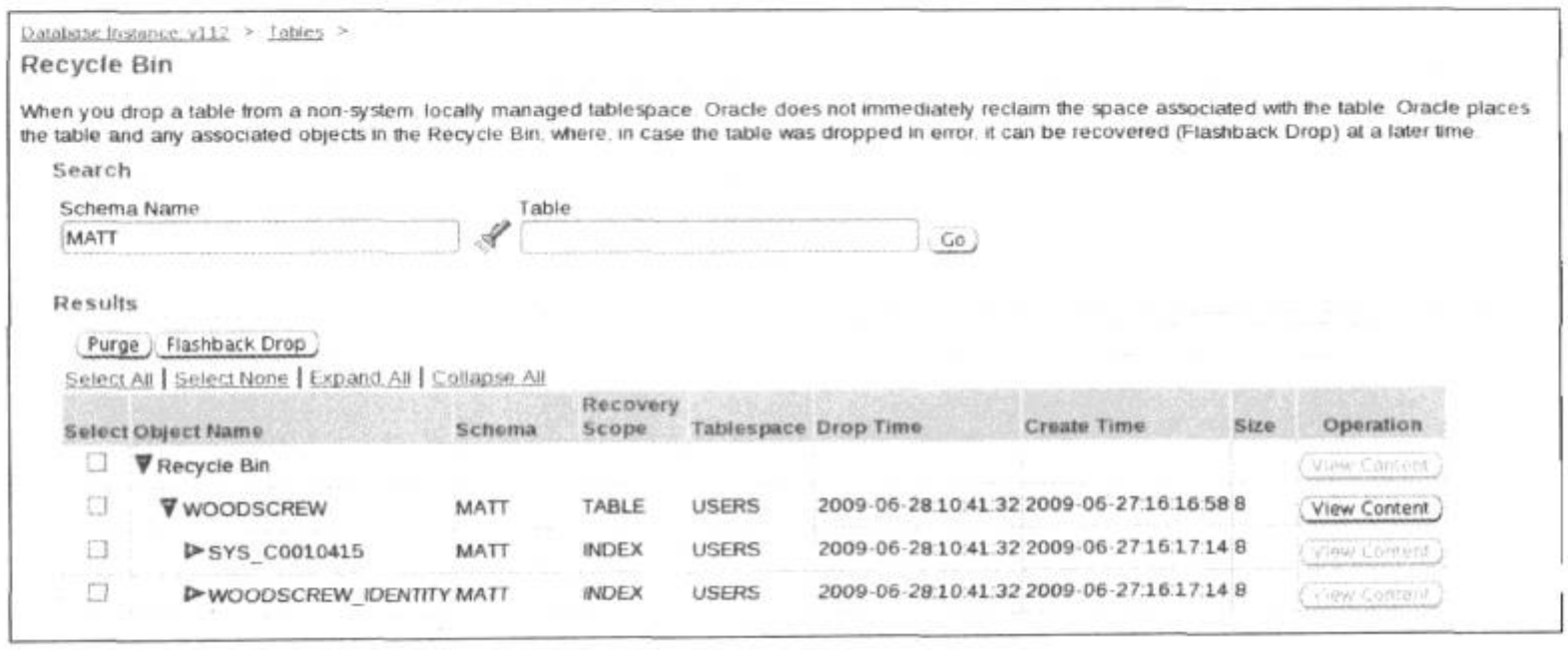


图 15-9　回收站

在该屏幕中需要注意的是，表 WOODSCREW 位于回收站中，主键索引和 WOODSCREW_IDENTITY 索引也在回收站中。同样需要注意的是右边的 View Content 按钮，该按钮用于查看已删除表中的行，从而确定它是否是将要闪回删除的对象。

步骤 3　选中表 WOODSCREW 后面的复选框，然后单击 Flashback Drop 按钮。

步骤 4　接下来的页面用于重命名撤消删除的对象。单击 Next 按钮后，可以从 OEM 中获得影响分析，该分析解释了将要闪回删除的具体对象，如图 15-10 所示。

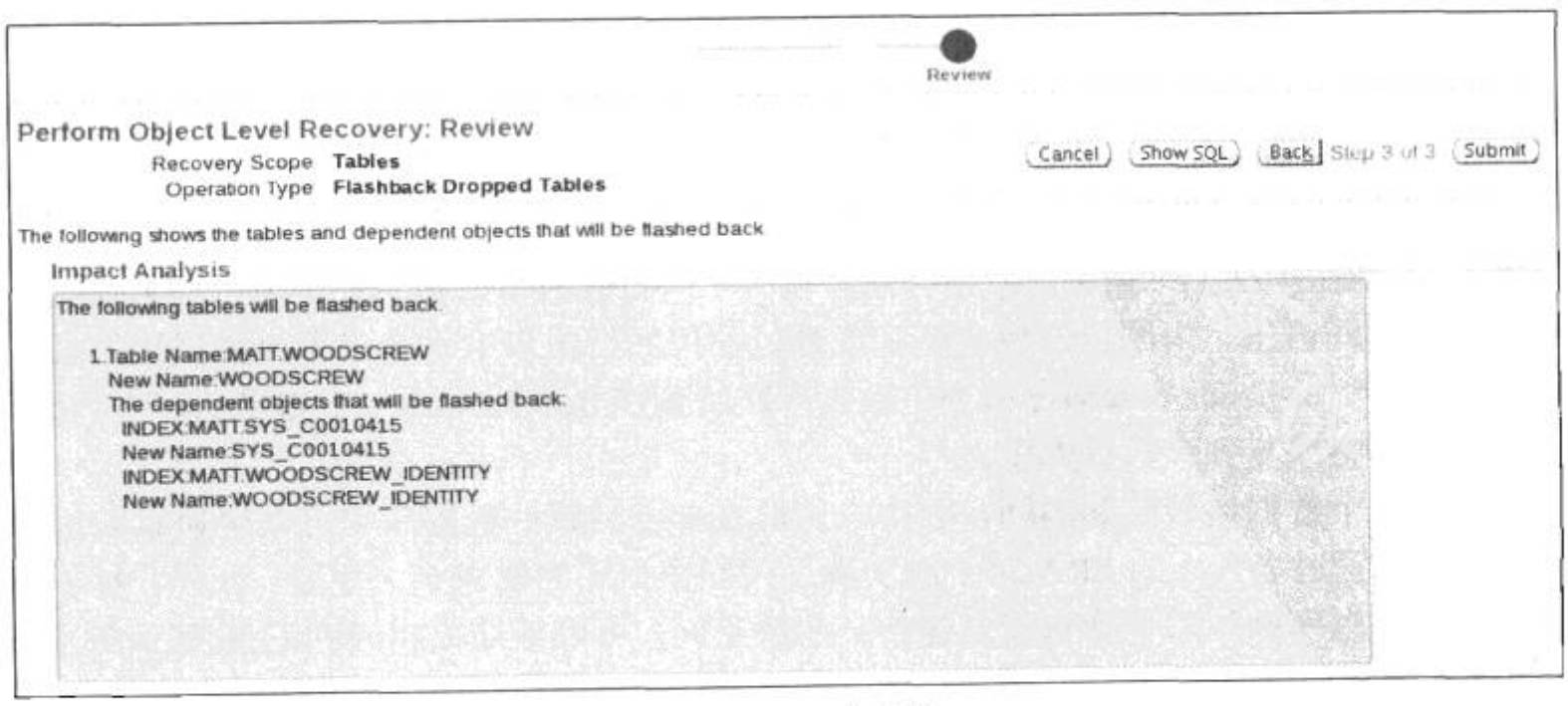


图 15-10　影响分析

步骤 5 单击 Submit 按钮。此时会显示一个确认屏幕，告诉我们所选择的表已经从回收站中还原。可以通过如下方法确认这一点：返回到 OEM 并选择 Administration | Schema | Tables 命令，并且选择 WOODSCREW 属主的模式。WOODSCREW 表将显示在原来的位置。

步骤 6 再次删除 WOODSCREW 表。单击 Recycle Bin 链接，可以再次看到已删除的表，如同步骤 2 所示。选择 WOODSCREW 后面的复选框，然后单击 Purge 按钮(Flashback Drop 旁边的按钮)。

步骤 7 此时显示一个确认屏幕，询问您是否希望清除所选择的对象和它们的相关对象。单击 Continue 按钮。WOOD 用户的回收站现在为空，不会显示 WOODSCREW 表和它的索引，已经从数据库中永久删除了这些对象。

15.6 闪回数据库

在 Oracle Database 11g 中，最具革命性的闪回技术可能也是最少使用的技术。闪回数据库提供了将整个数据库快速回绕到过去某个时间点的功能。该操作有如下操作：使用 RMAN 执行时间点恢复或执行用户管理的恢复实现的结果相同。然而，闪回数据库不需要从最新的备份中还原数据库的所有数据文件，并且不需要使用从最新的备份以来累积的所有归档日志执行前滚(roll-forward)操作。通过避免这些代价昂贵的操作，闪回数据库可以在很短的时间内执行时间点恢复。

闪回数据库的工作方式如下：以增量的方式记录指定时间间隔内已经更改的所有数据块。这些闪回“检查点”然后提供数据库可以“回绕”到的时间点。回滚到闪回检查点后，可以使用归档的日志前滚到通过 flashback database 命令指定的确切时间或 SCN。因此，该操作使用新的技术和旧的备用归档日志来提供执行时间点恢复的快速方法。

相比较而言，在闪回检查点后应用的归档日志比在最近备份(一般在每晚获得，而闪回日志每几分钟就获得一个备份)上应用的归档日志要少，因此闪回的恢复阶段非常快速。

15.6.1 闪回日志

闪回数据库实现了新的日志类型，称为闪回日志。闪回日志由数据库定期生成，并且在闪回恢复区(FRA)中累积。我们必须具有用于闪回数据库的 FRA；不可以在其他任何位置创建闪回日志。闪回日志包含从生成最近的闪回日志以来每个已更改数据块的映像副本。当执行 flashback database 命令将数据库回绕到在 flashback 命令中指定的时间点时，这些数据块将恢复到该数据库中。

由于整个数据块转储到闪回日志中，因此在非常活跃的数据库中可以快速累积这些数据块。设置适当大小的 FRA 对于成功满足闪回数据库的需求至关重要。此外，对于可以在闪回数据库操作后手动重新创建的某些表空间，可以手动关闭闪回记录(如下所示)，从而减少产生的日志数量。

```
alter tablespace ws_app_idx flashback off;
```

可以在任何时候重新打开闪回记录，如下所示。但需要注意的是，通过闪回记录间隙对关闭的表空间回绕时这样做是不值得的。

```
alter tablespace sales_idx flashback on;
```

对于通过 flashback database 命令关闭闪回记录任意一段时间的任何表空间，都需要在执行闪回数据库操作之前对其进行脱机操作。

15.6.2　闪回保存目标

闪回日志的生命周期与闪回数据库选项能恢复的时间范围直接相关。默认情况下，闪回日志保留足够长的时间，从而可以从当前时间闪回 24 小时。如果时间过长或过短，则可以使用初始化参数进行更改：

```
alter system set db_flashback_retention_target=720;
```

以分钟为单位指定该值(720 表示 12 小时)。

RMAN 实习：为闪回数据库操作进行配置

实习注意事项

该实习中将介绍最初配置数据库所需的基本步骤，这些配置步骤是为了将闪回记录用于闪回数据库操作。

步骤 1　关闭数据库并启动加载模式。数据库必须被加载，但是不打开：

```
SQL> select status from v$instance;
```

此外，检查以确保数据库处于 ARCHIVELOG 模式，闪回数据库操作需要数据库处于该模式：

```
SQL> archive log list;
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     62
Next log sequence to archive   64
Current log sequence           64
```

步骤 2　设置闪回保存目标为所需的值。我们将使用 12 小时作为窗口：

```
alter system set db_flashback_retention_target=720
SCOPE=BOTH SID='*';
```

步骤 3　设置 DB_RECOVERY_FILE_DEST 和 DB_RECOVERY_FILE_DEST_SIZE (FRA 参数)的值。需要注意的是，如果已经为 RMAN 备份策略设置了这些参数，则现在应该检查它们。闪回日志会在很大程度上增加 FRA 的使用率。我们需要将 FRA 的大小至少增加一倍。

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE = 2335825920
SCOPE=BOTH SID='*';
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST = '/u02/fra/'
SCOPE=BOTH SID='*';
```

步骤 4 打开闪回记录功能。使用与打开 ARCHIVELOG 模式相同的方式完成这项工作——当加载但不打开数据库时使用 alter database 命令:

```
alter database flashback on;
```

步骤 5 对于不需要的表空间，关闭闪回记录功能:

```
alter tablespace sales_idx flashback off;
```

步骤 6 打开数据库:

```
alter database open;
```

15.6.3 闪回数据库的调整

现在我们已经确定闪回数据库提供了所需的数据库回退位置，并且确定了希望回退到的时间点。我们已经设置了 DB_FLASHBACK_RETENTION_TARGET。现在的问题是:“如何知道 FRA 中是否有足够的空间来处理生成的大量闪回日志？从而断定执行了多少次闪回记录操作？”。下面的小节将给出这些问题的答案。

1. 使用 V$FLASHBACK_DATABASE_LOG

我们依次回答上面的问题。首先，Oracle 提供了内置的分析，用于确定是否需要增加 FRA 的大小。启用闪回记录后，Oracle 开始跟踪闪回记录的数量，并且将其存储在视图 V$FLASHBACK_DATABASE_LOG 中。该视图实际地提供了总闪回大小的估计:

```
select estimated_flashback_size from v$flashback_database_log;
```

需要注意的是，该视图给出了闪回日志的大小，但没有提供给 FRA 中的所有用户，因此需要将该值添加到所需的归档日志大小和 RMAN 备份大小。经过一段时间，这种估计的值会更加精确，这意味着当数据库执行完每日(然后是每月)的操作后，Oracle 就可以提供对闪回日志大小的更好估计。因此，较好的方法是再次检查这个估计量以查找是否具有正确的规范。

给定 FRA 的当前大小和当前可用的闪回日志，V$FLASHBACK_DATABASE_LOG 也可以提供将数据库闪回到的实际最早时间。这个时间也将显示 FRA 中的空间问题。下面的 select 语句将提供对闪回日志状态的基本理解:

```
select oldest_flashback_scn, oldest_flashback_time
from v$flashback_database_log;
```

2. 使用 V$FLASHBACK_DATABASE_STAT

Oracle 构建了一个监控视图，通过该视图可以查看闪回记录活动。V$FLASHBACK_

DATABASE_STAT 提供了在一段时间内生成的闪回数据的相关信息(一般是从 sysdate 向回扩展一小时的窗口)。除了显示闪回记录的数量，该视图也提交生成的重做和相同时间周期内生成的实际数据库数据。下面的 select 语句显示了该视图的示例输出：

```
select * from v$flashback_database_stat;
BEGIN_TIM END_TIME FLASHBACK_DATA DB_DATA REDO_DATA
--------- --------- -------------- ---------- ----------
ESTIMATED_FLASHBACK_SIZE
------------------------
19-SEP-05 19-SEP-05       9003008    15294464    3986944
  210247680
19-SEP-05 19-SEP-05      15884288    21225472    5766144
  210247680
19-SEP-05 19-SEP-05      10248192    25772032    5162496
  210075648
```

RMAN 实习：执行闪回数据库

实习注意事项

在该实习中将对闪回数据库进行实际操作。我们将引入一个不可以通过任何其他的、较少干扰的闪回进行处理的错误：表截断。由于截断不产生任何重做，因此不可以执行闪回表操作。因此，我们必须执行整个数据库的闪回。

在这里，我们使用在前面的 RMAN 实习“闪回版本查询”中构建的 WOOD.WOODSCREW_INVENTORY 表。如果没有构建该表，现在就构建它。

步骤 1　首先，从数据库中获得当前的 SCN。因为只是进行测试，所以可以通过如下方法为测试做准备：在数据库中添加错误之前获得当前的 SCN。

```
SQL> select current_scn from v$database;
CURRENT_SCN
-----------
885524
```

步骤 2　引入错误：

```
SQL> truncate table wood.woodscrew_inventory;
Table truncated.
```

步骤 3　关闭数据库，然后重新加载。数据库必须加载且没有打开以进行闪回操作。

```
SQL> connect / as sysdba
Connected.
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
```

步骤 4　执行 flashback 命令：

```
SQL> flashback database to scn 885524;
```

```
Flashback complete.
```

步骤 5 以只读模式打开数据库，确认表已经闪回到适当的 SCN:

```
SQL> alter database open read only;
Database altered.
SQL> connect wood/wood;
Connected.
SQL> select count(*) from woodscrew_inventory;
  COUNT(*)
----------
         4
```

步骤 6 使用 resetlogs 操作打开数据库:

```
SQL> connect / as sysdba
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter database open resetlogs;
```

15.7 闪回数据归档(完全撤消)

在 Oracle 11g 中提供了一个新的闪回功能——闪回数据归档(有时称作 Oracle 完全撤消(Total Recall))，闪回数据归档(Flashback Data Archive)是一组功能的集合，使用闪回数据归档可以把所有更改永久归档到一个表中，以便及时回滚到任一时间点和查看过去的数据。和闪回查询不同的是，闪回查询依赖于撤消的短暂性，而闪回数据归档要求构建一个特定类型的表空间，以便保存要求在特定表中及时回滚的版本信息。

闪回数据归档非常有助于审核和归档那些对法律或法规敏感的表。可以配置一个设定保留期的数据归档，该保留期匹配商业需要或法规需要，然后将表分配给归档。一旦完成操作，就可在 SQL 中利用一个简单的闪回查询来及时回退到表中几个月甚至几年前的时间点。

应该基于归档数据怎么生成和访问来记住这两个体系结构的约束。如果将表置于数据归档模式，将不能进行下列操作：

- 执行带有 UPGRADE 子句的 alter table 命令
- 执行删除表操作

上面操作均会产生如下错误：

```
ORA-55610:Invalid DDL statement on history-tracked table
```

RMAN 实习：创建一个闪回数据归档

实习注意事项

在该实习中我们将创建一个闪回数据归档，然后将它添加 WOODSCREW_INVENTORY 表到归档中。该数据归档配额大小规定为 500MB，保留期为一年。

步骤 1 为闪回数据归档创建一个表空间。

```
SQL> create tablespace FBS datafile '/home/oracle/app/oracle/oradata/
v112/fbs01.dbf' size 1024M;
```

步骤 2 在 FBS 表空间中创建一个数据归档。

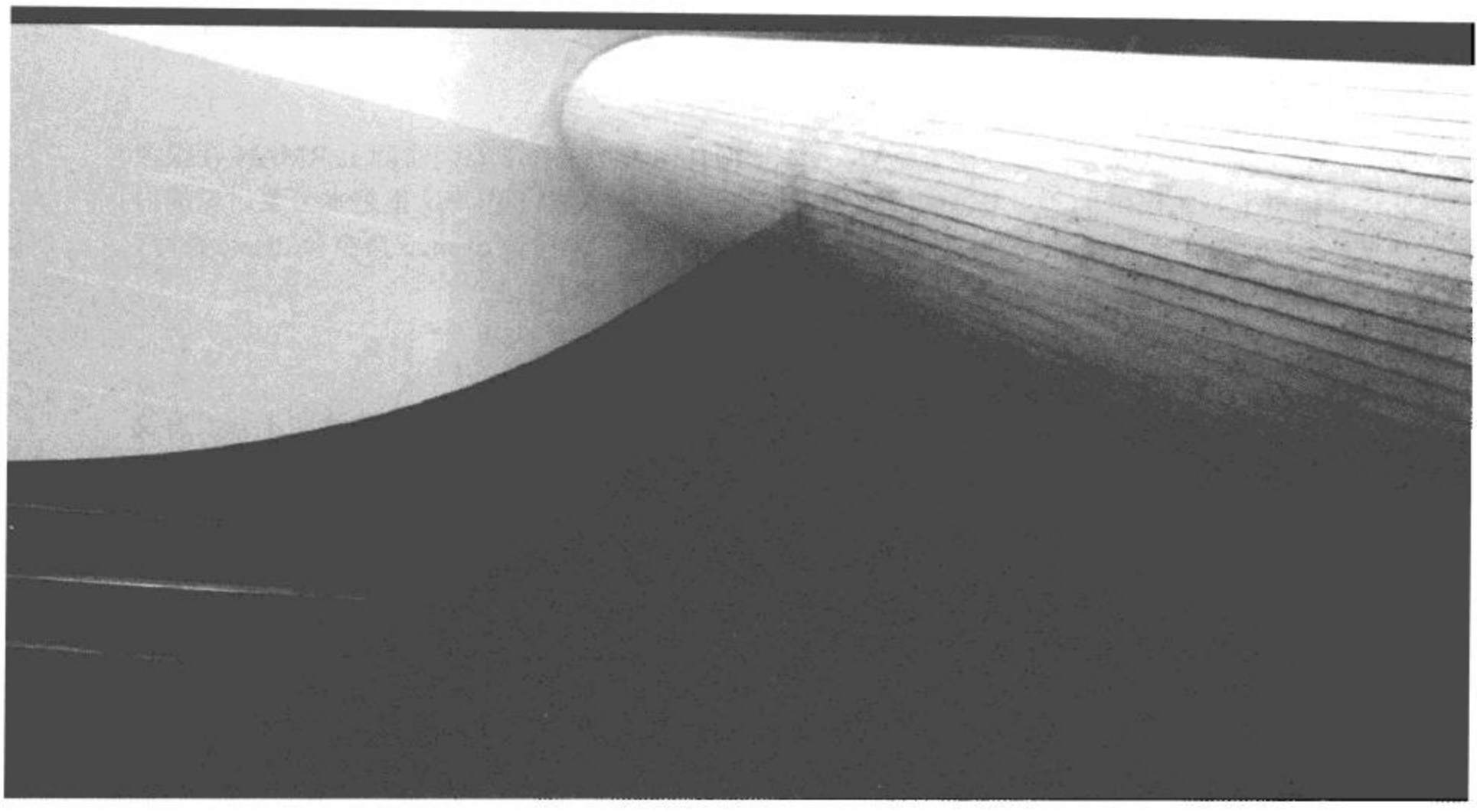
```
SQL>create flashback archive default FDA01 tablespace FBS quota 500M
retention 1 year;
```

步骤 3 将 matt.woodscrew_inventory 表添加到闪回数据归档 FDA01 中。这很简单，因为没有特别说明，对于任何要添加的表，其默认数据归档为 FDA01。

```
SQL> alter table matt.woodscrew_inventory flashback archive;
```

15.8 小结

本章首先介绍了从用户导致的错误中恢复的新方法，该方法在 Oracle Database 中称为闪回技术。相比于彻底的介质恢复，新的闪回技术允许我们以较快的、影响较少的方式从逻辑错误中恢复。我们讨论了使用闪回版本查询和闪回事务查询来确定逻辑损坏涉及的所有范围，也说明了使用闪回表命令从错误的 DML 语句和表删除中恢复的方法。接着讨论了 Oracle Database 11g 新的闪回事务选项，该选项利用重做日志撤消事务级错误操作而不是表级错误操作。同时还讨论了新的闪回数据库功能，使用该功能可执行整个数据库的时间点恢复，而不需要从备份中完全还原数据文件。本章最后回顾了 11g 中新增的功能——创建闪回数据归档，该功能可以在存储记录的空间中回退到任一时间点查看过去的数据。



第16章

RMAN 维护

熵(entropy)是热力学中的一个难题。从本质上说，熵描述了系统从有序变为无序的趋势，这个趋势的结果就需要维护系统。RMAN 也是如此。使用 RMAN 时，我们必须维护许多事件以保证 RMAN 的平稳运行。

本章将介绍 RMAN 的维护。从 crosscheck 命令到保留策略和冗余策略，这里将介绍您需要知道的、与防止 RMAN 崩溃有关的所有问题。讨论完 RMAN 的维护问题后，我们还将讨论与恢复目录有关的一些问题，包括对恢复目录模式的介绍。

16.1　RMAN 的维护

您不会认为只需要使用 RMAN 而不用做任何维护工作吧？但实际上 RMAN 在很大程度上是不用维护的，不过您应该知道一些与维护有关的工作，这正是本节要讨论的内容。我们首先介绍 crosscheck 命令以及保留策略，接下来讨论 change 命令和 delete 命令，最后还要讨论如何在 RMAN 中为现有的数据库备份编写目录。

16.1.1　交叉校验 RMAN 备份

您可能会遇到这样的情况：控制文件或恢复目录中列出的备份集片或副本并没有物理存在于备份介质(磁盘或磁带)上。构成这些备份或副本的物理文件可能会被某些进程(如用于磁带管理系统的独立保留策略)删除或因磁带损坏而删除，此外备份集片所在设备的物理故障也会导致文件实际上已丢失。

在 RMAN 目录和物理备份目的地不同步的情况下，我们可以使用 crosscheck 命令来校验控制文件或恢复目录中的 RMAN 信息是否与备份介质上的实际物理备份集片相同。

使用 crosscheck 命令时，我们关心的是每个备份集或副本的状态。如果使用控制文件，用于备份集片的 V$BACKUP_SET 视图和用于副本的 V$DATAFILE_COPY 视图中的 STATUS 列列出了每个备份集或副本的状态码；如果使用恢复目录，则在备份集片的 RC_BACKUP_SET 和副本的 RC_DATAFILE_COPY 中列出了每个备份集或副本的状态码。在不同的备份状态码中，我们主要关心下面两种状态码：

- A(AVAILABLE，可用)：RMAN 认定该项存在于备份介质上。
- X(EXPIRED，不可用)：这个备份集片或副本存储在 RMAN 目录(即控制文件或恢复目录)中，但是并没有物理存在于备份介质上。

执行 crosscheck 命令时，RMAN 检查目录中列出的每个备份集或副本并且判断它们是否存在于备份介质上。如果备份集或副本不存在于备份介质上，它就会被标记为 EXPIRED，并且不能用于任何还原操作；如果备份集或副本存在于备份介质上，它就会维持 AVAILABLE 状态。如果以前被标记为 EXPIRED 的备份集或副本再次存在于备份介质上(例如恢复了损坏的磁盘驱动器之后)，crosscheck 命令就会将它标记回 AVAILABLE。

crosscheck 命令必须连接到目标数据库上运行，而不用连接到恢复目录。如果您的备份存储在磁盘上，那么在发送 crosscheck 命令之前不必分配信道。如果将备份存储在磁带上(例如，通过 MML)，那么在发送 crosscheck 命令之前必须分配维护信道。如果预先已经配置了信道，那么就不需要做这些工作了。如果您用混合信道(例如，一些针对磁盘，一些针对 SBT)，那么 crosscheck 命令仅检查这些备份，这些备份在备份的过程中利用信道创建而成。

在下面的示例中，我们执行 crosscheck 命令来检查所有备份集的状态，并且判断它们是否存在于备份介质上。

```
RMAN> crosscheck backup;
```

```
using channel ORA_DISK_1
crosschecked backup piece: found to be 'AVAILABLE'
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/01_mf_annnn_TAG20090908T025101_5bd6qydw_.bkp RECID=2 STAMP=696999070
crosschecked backup piece: found to be 'AVAILABLE'
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/O1_mf_nnndf_TAG20090908T025112_5bd6r0mz_.bkp RECID=3 STAMP=696999072

crosschecked backup piece: found to be 'EXPIRED'
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
o1_mf_annnn_TAG20090908T031112_5bd7xjo7_.bkp
RECID=9 STAMP=679000272

Crosschecked 3 bjects
```

在这个示例中，我们交叉校验了全部 4 个备份集片。可以看到，crosscheck 命令的输出分为两部分，第一部分列出确定存在于备份介质上的所有备份集片；第二部分列出确定不存在于备份介质上的所有备份集片，因此 RMAN 将这些备份集片设置为 EXPIRED 状态。需要注意的是，crosscheck 命令不会将备份集片的状态从 DELETED 改为 AVAILABLE，任何标记为 DELETED 状态的备份都不能被修改。

我们还可以交叉校验数据文件备份、表空间备份、控制文件备份以及服务器参数文件备份。此外，可以通过识别与备份关联的标记来选择要交叉校验的特定备份。基于使用的设备或者基于一个时间周期，我们甚至还可以交叉校验所有备份。下面是一些交叉校验备份的示例：

```
crosscheck backup of datafile 1;
crosscheck backup of tablespace users;
crosscheck backup of controlfile;
crosscheck backup of spfile;
crosscheck backup tag='SAT_BACKUP';
crosscheck backup completed after 'sysdate - 2';
crosscheck backup completed between 'sysdate - 5' and 'sysdate - 2;
crosscheck backup device type sbt;
```

上面的第一个示例是备份的交叉校验。当然，也可以交叉校验归档日志备份，如下所示：

```
RMAN> crosscheck archivelog all;
released channel:ORA_DISK_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=36 device type=DISK
validation succeeded for archived log
archived log file
name=/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
```

```
/01_mf_1_7_5bd8qq6g_.arc RECID=10 STAMP=697001111
validation succeeded for archived log
archived log file
name=/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
01_mf_1_8_5bd8qr29_.arc RECID=11 STAMP=697001112
validation failed for archived log
archive log filename=/oracle/app/oracle/flash_recovery_area/ROB1/
archivelog/2009_09_08
/01_mf_1_9_5bd8qv45_.arc RECID=12 STAMP=697001115
validation succeeded for archived log
archive log file name=/oracle/app/oracle/flash_recovery_area/ROB1/
archievelog/2009_09_08
/01_mf_1_10_5bd8rco9_.arc RECID=13 STAMP=697001131

Crosschecked 4 objects
```

我们可以基于一些标准(包括时间、具体的或指定范围的 SCN 或日志序列号)来交叉校验归档的重做日志备份，甚至还可以使用 like 参数与通配符来交叉校验特定的归档日志备份。下面的示例说明了 crosscheck 命令中的一些变化：

```
crosscheck archivelog like 'ARC001.log';
crosscheck archivelog 'D:\ORACLE\ADMIN\RECOVER\ARCH\ARC00012.001';
crosscheck archivelog like '%ARC00012.001';
crosscheck archivelog from time "to_date('01-10-2006', 'mm-dd-yyyy')";
crosscheck archivelog until time "to_date('01-10-2006', 'mm-dd-yyyy')";
crosscheck archivelog from sequence 12;
crosscheck archivelog until sequence 522;
```

使用 crosscheck copy 命令可以交叉校验副本。我们可以交叉校验数据文件副本、控制文件副本、归档重做日志副本以及(磁盘上的)归档的重做日志。下面是两个交叉校验这些对象类别的示例：

```
crosscheck copy of datafile 5;
crosscheck datafilecopy 'd:\oracle\oradata\recover\recover_users_
1.dbf';
```

RMAN 实习：使用 crosscheck 命令

实习注意事项

本实习假定 Oracle 数据库已经运行在 ARCHIVELOG 模式中，此外还假定在磁盘上备份数据库，数据库中包含一个 USERS 表空间以及一个与 USERS 表空间关联的数据文件。注意我们的示例输出可能不完全同于您的输出。

步骤 1 使用 RMAN 备份 USERS 表空间。

```
RMAN> backup tablespace users;
Starting backup at 08-SEP-09
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=00004 name=/ora01/oracle/rob1/rob1/users01. dbf
channel ORA_DISK_1: starting piece 1 at 08-SEP-09
channel ORA_DISK_1: finished piece 1 at 08-SEP-09
piece handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/
2009_09_08/o1_mf_nnndf_TAG20090908T032848_5bd8yjmq_.bkp
tag=TAG20090908T032848 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 08-SEP-09
Starting Control File and SPFILE Autobackup at 08-SEP-09
piece handle=/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/
2009_09_08/o1_mf_s_697001330_5bd8ylgv_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 08-SEP-09
```

步骤 2　查看备份操作的输出，确定刚刚创建的这个备份集片。在步骤 1 的输出中突出显示了这个备份集片。注意我们没有删除控制文件自动备份集片。

步骤 3　从磁盘中删除这个备份集片。

步骤 4　执行 crosscheck 命令来确定这个备份集片的状态。RMAN 会检测出这个备份集片已被删除并且将它标记为 EXPIRED 状态。注意，控制文件自动备份片仍然可用。

```
RMAN> crosscheck backup;
using channel ORA_DISK_1
crosschecked backup piece: found to be 'AVAILABLE'
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
o1_mf_annnn_TAG20090908T025101_5bd6qydw_.bkp
RECID=2 STAMP=696999070
crosschecked backup piece: found to be 'AVAILABLE'
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
o1_mf_nnndf_TAG20090908T025112_5bd6r0mz_.bkp
RECID=3 STAMP=696999072
crosschecked backup piece: found to be 'EXPIRED'
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
o1_mf_annnn_TAG20090908T031112_5bd7xj07_.bkp
RECID=9 STAMP=697000272
crosschecked backup piece: found to be 'EXPIRED'
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08/
o1_mf_nnndf_TAG20090908T032848_5bd8yjmq_.bkp
RECID=14 STAMP=697001328
crosschecked backup piece: found to be 'AVAILABLE'
backup piece
handle=oracle/app/oracle/flash_recovery_area/ROB1/autoback/2009_09_08/
o1_mf_s_697001330_5bd8ylgv_.bkp
RECID=15 STAMP=697001330
Crosschecked 5 objects
```

16.1.2　RMAN 备份的验证

RMAN 提供的 validate 命令允许查看给定的备份集和进行验证以确保这个备份集能够被还原。validate 命令与 backup validate 命令相似，带有很少的附加修饰。我们可以使用 validate 命令选择要验证的备份集，而 backup 命令或 restore 命令的 validate 参数则允许 RMAN 确定要验证的备份集。

一般来说，在生产的环境下，当完成备份时可以使用 backup validate check logical 命令执行数据库的验证。因此，validate 命令常用在检查特定备份集或者在闪回恢复区中检查所有备份集的特殊情况下。

validate 命令不会检查所有的数据块。那些从没使用过的数据块将不会被检查。如果正在运行 10.2 或者更大的兼容集时，那么以前使用过但现在没使用过的本地管理的表空间包含的块将不会被验证。

validate 命令只默认检查物理损坏，而逻辑损坏不以考虑。如果同时也想检查块的逻辑损坏，则使用 validate 命令的 check logical 选项。在 validate 命令完成检查后，它将会填充在 V$DATABASE_BLOCK_CORRUPTION 视图中。如果已经配置了数据库监视器，那么在监视器视图中包含行数量的计数是很好的方法。这种方式能判断是否存在需要处理的损坏问题。

validate 命令不会发现某种损坏类型。因此，运行这些命令后可能数据库仍然存在问题，虽然这种情况极少，但是仍然存在。一个例子就是块与块之间的损坏问题，这种损坏发生在两个块之间，例如在块链接的情况下。validate 命令不会检测块与块之间的损坏。

接下来将介绍使用 validate 命令获得备份集的主键并验证该备份集。需要注意的是，必须向该命令传递要验证的备份集的主键 ID，这个主键 ID 可以通过运行 list backupset summary 命令来确定。

```
RMAN>list backup summary;
List of Backups
===============
Key  TY LV S Device  Type Completion  Time #Pieces #Copies Compressed Tag
------- -- -- - ---------- ------------ ------- ------- ---------- ---
2    B  A  A  DISK       08-SEP-09      1       1       YES TAG200908T025101
3    B  F  A  DISK       08-SEP-09      1       1       YES TAG200908T025112
4    B  F  A  DISK       08-SEP-09      1       1       YES TAG200908T025112
5    B  A  A  DISK       08-SEP-09      1       1       YES TAG200908T025311
6    B  A  A  DISK       08-SEP-09      1       1       NO  TAG200908T025326
7    B  F  A  DISK       08-SEP-09      1       1       NO  TAG200908T025328
8    B  A  A  DISK       08-SEP-09      1       1       NO  TAG200908T025355
9    B  A  X  DISK       08-SEP-09      1       1       YES TAG200908T031112
10   B  F  X  DISK       08-SEP-09      1       1       YES TAG200908T031114
11   B  A  X  DISK       08-SEP-09      1       1       YES TAG200908T031116
12   B  A  A  DISK       08-SEP-09      1       1       NO  TAG200908T032531
13   B  F  A  DISK       08-SEP-09      1       1       NO  TAG200908T032815
14   B  F  X  DISK       08-SEP-09      1       1       NO  TAG200908T032848
15   B  F  A  DISK       08-SEP-09      1       1       NO  TAG200908T032850
```

注意不同的列，这些列在 Oracle Database Backup 和 Rocovery Reference Guide(存在许多的列)中定义。在这种情况下，大多数列都是自我解释的。除了以下内容：

- Key　每个备份都有唯一的 Key 标识符。
- TY　该类备份：或者是备份集(B)或者是代理副本(P)。
- LV　指定了备份的类型，完整的数据库备份集都用 F 标记。归档重做日志备份用 A 来标记。如果这个备份是增量备份，那么 LV 列将包含一个 0 针对基础备份或者是 1 针对一个增量备份。
- S　指定了备份的状态，这个值包含 A(可用)，U(不可用)和 X(如果所有的备份集片已经过期)。

为保证这个备份是可用的(没有损坏，相对于物理存在)。可以验证这个备份集。在这种情况下，在该示例中是归档日志备份：

```
RMAN> validate backupset 2;
```

我们也可以在闪回恢复区中验证所有备份：

```
RMAN> validate recovery area;
```

16.1.3　备份保留策略

第 3 章介绍过 RMAN 备份和副本的保留策略的配置。保留策略(retention policy)是管理备份与副本以及指定它们在备份介质上的保留时间的一种方法。我们可以定义两种基本的保留策略类别：恢复窗口备份保留策略(recovery window backup retention policy)和备份冗余备份保留策略(backup redundancy backup retention policy)。下面将分别讨论。

每个冗余策略都将一直保持直到被修改或被删除(或者直到使用 create controlfile 命令重新构建控制文件的时候)。两个冗余策略是相互排斥的。此外，即使使用了冗余策略，也只能在使用带 obsolete 参数的 delete 命令的时候才能删除物理备份。

下面我们将详细讨论这两种备份保留策略类型，然后再介绍如何管理备份策略标记为过时的 RMAN 备份和副本。

1. 恢复窗口备份保留策略

这种保留策略是基于想要数据库恢复到的最近日期定制的。如果要将数据库恢复到两星期前的某个时间点，就可以使用恢复窗口备份保留策略让 Oracle 保存那些能够确保我们恢复操作成功的备份。

例如，假设今天是星期一，此时存在 3 个备份。第一个备份是在昨天(星期天)生成的，第二个备份是在上一个星期四生成的，而最后一个备份是在十天前(再上一个星期六)生成的。如果恢复窗口被设置为 7 天，前两个备份(星期天和上一个星期四生成的)会被视为当前的备份，而十天前生成的备份(再上一个星期六生成的)会被认为已过时。如果要为 RMAN 建立这个 7 天的冗余策略，则可以使用带有 retention policy to recovery window 参数的 configure 命令，如下所示。

```
configure retention policy to recovery window of 7 days;
```

2. 备份冗余备份保留策略

使用备份冗余备份保留策略时，RMAN 会从最新备份开始保留 x 个数据库备份。例如，假设我们配置的备份冗余为 3，同时又在星期一、星期二、星期三和星期四执行备份操作。因为保留策略被设置为 3，所以只要星期四的备份操作成功，Oracle 就会丢弃星期一生成的备份。

使用带有 retention policy to redundancy 参数的 configure 命令可以启用备份窗口备份保留策略。可以使用下面的命令将备份窗口备份保留策略设置为 3：

```
configure retention policy to redundancy 3;
```

3. 保留策略的维护

当一个给定的备份或副本满足备份保留策略的标准而被丢弃的时候，具体结果取决于您是使用闪回恢复区(FRA)还是手动备份区的保存策略。我们可以修改备份使其备份保留策略与默认保留策略不同。让我们看看下面的这些选项。

使用闪回恢复区的保留策略维护 如果使用闪回恢复区，RMAN 最终会基于空间需求从 FRA 中删除备份集片，图像副本或归档重做日志(我们目前只访问这些备份)。这意味着备份仍然占用一定的空间和时间，直到其他备份或归档日志需要时才释放空间。结果，可能会观察 FRA 的空间使用情况，因为当感觉空间是稳定时它是在增长的。可以在查询 V$RECOVER_FILE_DEST 视图的 SPACE_RECLAIMABLE 的列时确定过时的备份所占用的 FRA 空间的数量。利用这个视图，能确定是否已经为 FRA 分配充足的空间。以下是一个查询 V$RECOVER_FILE_DEST 视图的示例：

```
SQL>select * from v$recovery_file_dest;
NAME                                 SPACE          SPACE       SPACE       NUM
                                     LIMIT          USED   RECLAIMABLE  FILES
-------------------------------------------------------------------------------
c:\oracle\flash_recovery_area 3,221,225,472 934,699,008 23,445,504 17
```

在这种情况下，我们发现大约 23MB 的空间是可以回收的。一旦需要可用空间时，RMAN 将清除这些文件。如果我们发现使用过的空间数量非常多同时可回收空间数量非常少，那么我们可能给 FRA 分配了过少的空间。如果我们发现可回收的空间数量很多，那么可以检查您的保留标准并且确保他们是设置正确的，并且确保没对 FRA 过量的分配空间。

如果发现为已过时的备份集分配了空间，则可以使用 RMAN 的 report obsolete 命令来查看该备份集的详细信息，如下所示：

```
RMAN>report obsolete;
        using target database control file instead of recovery catalog
        RMAN retention policy will be applied to the command
        RMAN retention policy is set to redundancy 2
        Report of obsolete backups and copies
```

```
Type                 Key    Completion Time   Filename/Handle
-------------------- -----  ---------------   ---------------
Backup Set           61     08-SEP-09
Backup Piece         71     08-SEP_09

/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697001330_5bd8ylgv_.bkp
```

使用手动备份区的保留策略维护　如果没有使用 FRA，则将备份到手动备份区。在这种情况下需要有区别地管理已过期的备份。与 FRA 一样，一旦不再需要备份，RMAN 将标记备份或副本为 OBSOLETE。使用 report obsolete 命令能确定哪个备份 RMAN 已经被标记为 OBSOLETE。(我们在下一章将介绍 report 命令。)

```
report obsolete;
```

在手动备份区的例子中，RMAN 删除已过时的备份需要执行更多的步骤。额外的步骤就是 delete obsolete 命令的执行。该命令要求 RMAN 在物理磁盘上物理的删除备份。它也标记控制文件中删除备份时的相关在元数据，并且 delete obsolete 命令将删除恢复目录上与备份相关联的所有记录。

使用闪回恢复区保留策略维护　当然在查看备份时，也可能需要更改它的保留策略。在这种情况下，可以使用带有 keep 参数的 RMAN 的 change 命令修改需要保留的备份的保留策略。使用这个命令时，其保留策略被修改的备份(或副本)会被认为是一个长期备份(long-term backup)，且不受相应的保留策略的影响。

在 Oracle Database 11g，可以使用 change 命令将一个备份修改为归档备份(第 11 章介绍过)。这个归档备份的结果将永久保留(这需要恢复目录)，或者在归档备份将被 RMAN 标记为过时后定义新的数据(如果保留的数据时间超过一年，就需要用恢复目录)。在归档备份中，相关的归档重做日志备份也被保留并且在指定的时间内标记为 obsolete (假设他们不需要其他的归档备份)。下面是使用 Oracle Database 11g 的 change 命令创建一个归档备份的示例：

注意：
如果使用闪回恢复区，就不可以使用 keep 命令的 keep forever 子句。

```
-- Run the list backup command to get the backup key we need
-- This is just a partial listing of the list backup output
-- We bolded the backup set key values that we will be using.
RMAN> list backup;
List of Backup Sets
===================
BS Key Type LV Size         Device Type Elapsed Time Completion Time
------- ----------          ----------- ------------ ---------------
267     Full    1.09G       DISK        00:06:50     12-FEB-10
        BP Key: 200     Status: AVAILABLE Compressed: NO Tag: TAG20100212
        T121310
```

```
Piece Name: C:\ORACLE\FLASH_RECOVERY_AREA\CLASS\BACKUPSET\2010_02_12\
O1_MF_NNNDF_TAG20100212T121310_5QCB274D_.BKP
  List of Datafiles in backup set 3
-- change a backup so it will be retained forever. 267 is the backup key
-- that we get from the list backup RMAN command
-- Change backup 267 from an obsolete status to a keep status.
-- Note that this command requires a recovery catalog
change backupset 267 keep forever;

-- change a backup so it will be kept for 7 more days
change backupset 267 keep until time 'sysdate + 7' logs;

-- Obsolete a backup
change backupset 33 nokeep;
```

在 Oracle Database 11g 以前的 Oracle 版本中，change 命令修改以前声明的保留标准。这些虽然不需要考虑归档备份，但结果归档重做日志备份不能自动保留。为确保备份总是可及时恢复到当前时间点上(但考虑存储的影响)，需要使用 keep logs 参数确保归档重做日志备份的保存标准被修改，如下例所示：

```
change backupset 267 keep forever logs;
```

修改一个备份或副本时，必须引用与这个备份关联的备份集键。从示例中可以看出，可以使用 list backup 或 list copy 命令来查找备份集键(正如现在的情况，备份集键是 267，在第 17 章将介绍这些命令)。注意 keep forever 选项有两个重要的限制。首先，只有在使用任何版本的恢复目录时才可使用该选项。其次，如果使用 Oracle Database 11g 版本之前的 FRA，则不可以使用该选项。

检查完报告后，可以使用带有 obsolete 参数的 delete 命令指示 RMAN 删除过时的备份，如下所示：

```
delete obsolete;
```

16.1.4 change 命令

在本章的前面部分已介绍过 change 命令，并说明了如何使用这个命令来修改为特定备份指定的保存窗口。change 命令允许用户修改备份的状态。我们可能会遇到备份介质设备在某个时间段中无效的情况(可能是因为某人切断了该设备的电源)。在这种情况下，可以使用 change 命令来指示这个设备上的备份是不可用的。

解决硬件故障和修复磁盘后，可以再次执行 change 命令将备份改为可用(AVAILABLE)的状态。也可以将备份修改为不可用(UNAVAILABLE)的状态，以指示该备份当前是不可用的。在还原与恢复操作期间，不会考虑那些不可用的备份，不过在执行 delete expired 命令期间这些备份记录不会被删除。下面是一些 change 命令的使用示例。

```
change backup of database tag = 'GOLD' unavailable;
change backup of database like '%GOLD%' unavailable;
```

```
change backupset 33 unavailable;
change backupset 33 available;
change archivelog 'd:\oracle\mydb\arch\arch_001.arc' unavailable;
```

可以使用 change 命令修改归档的重做日志备份的状态。例如，可以将已备份了指定次数的所有归档的重做日志备份修改为不可用的状态，还可以修改特定设备上的所有备份的状态。下面是这些操作的示例：

```
change archivelog all backed up 5 times to device type disk unavailable;
change backup of database device type disk unavailable;
```

还可以使用带有 delete 参数的 change 命令删除备份集(即在备份介质上的物理删除，并且从控制文件和恢复目录中删除)。首先，必须使用 list backup 或 list copy 命令(第 17 章将详细介绍这些命令)标识要删除备份的 RMAN 备份 ID。下面是一个使用 list backup 命令获取备份 ID 的示例(由于篇幅关系，我们对输出做了一些修改以节省空间)：

```
RMAN> list backup;
List of Backup Sets
===================
BS Key   Size
------- ----------
56        18.88M

List of Archived Logs in backup set 56
Thrd Seq     Low SCN    Low Time  Next SCN   Next Time
---- ------- ---------- --------- ---------- ---------
1    69      593930     07-SEP-09  612201    07-SEP-09
1    70      612201     07-SEP-09  635114    07-SEP-09
1    71      635114     08-SEP-09  635383    08-SEP-09
Backup Set Copy #1 of backup set 56
Device Type Elapsed Time Completion Time  Compressed    Tag
----------- ------------ -------------    ----------    -----
DISK        00:00:05     08-SEP-09        YES       TAG20090908T025101

List of Backup Pieces for backup set 56 Copy #1
BP Key   Pc# Status     Piece Name
-----    ---  --------   --------
72        1    AVAILABLE
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd6qp1q_.bkp
Backup Set Copy #2 of backup set 56

Device Type Elapsed Time Completion Time Compressed Tag
----------- ------------ -------------   --------- ----
DISK        00:00:05     08-SEP-09        YES       TAG20090908T025101
List of Backup Pieces for backup set 56 Copy #2
BP Key   Pc# Status     Piece Name
-----    ---  --------   --------
87        1    AVAILABLE
```

```
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd9od5x_.bkp

BS Key   Size
------- ----------
57       57.50K

List of Archived Logs in backup set 57
Thrd Seq     Low SCN     Low Time  Next SCN   Next Time
---- ------- ---------- --------- ---------- ---------
1    1       635384     08-SEP-09 635574     08-SEP-09


Backup Set Copy #1 of backup set 57
Device Type Elapsed Time Completion Time Compressed Tag
---------- ------------ --------------- ---------- ----
DISK        00:00:00     08-SEP-09       YES        TAG20090908T025101

List of Backup Pieces for backup set 57 Copy #1
BP Key   Pc# Status     Piece Name
------   ---  --------  --------
73       1    AVAILABLE
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd6qydw_.bkp
Backup Set Copy #2 of backup set 57
Device Type Elapsed Time Completion Time Compressed Tag
XX
---------- ------------ --------------- ---------- ----
DISK        00:00:00     08-SEP-09       YES        TAG20090908T025101

List of Backup Pieces for backup set 57 Copy #2
BP Key   Pc# Status     Piece Name
------   ---  --------  --------
88       1    AVAILABLE
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd9ohdb_.bkp

BS Key   Type  LV   Size
------   ----  ---  ----
58       Full  243.80M
List of Datafiles in backup set 58
File Lv Type Ckp SCN   Ckp Time Name
---- -- ---- --------  -------- ----
1    Full    635590    08-SEP-09 /ora01/oracle/rob1/rob1/system01.dbf
2    Full    635590    08-SEP-09 /ora01/oracle/rob1/rob1/sysaux01.dbf
3    Full    635590    08-SEP-09 /ora01/oracle/rob1/rob1/undotbs01.dbf
4    Full    635590    08-SEP-09 /ora01/oracle/rob1/rob1/users01.dbf
Backup Set Copy #1 of backup set 58
Device Type Elapsed Time Completion Time Compressed Tag
---------- ------------ --------------- ---------- ----
```

```
DISK        00:01:53     08-SEP-09       YES       TAG20090908T025112

List of Backup Pieces for backup set 58 Copy #1
BP Key   Pc# Status     Piece Name
-------  ---  --------  --------
74       1    AVAILABLE
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnndf_TAG20090908T025112_5bd6r0mz_.bkp

Backup Set Copy #2 of backup set 58
Device Type Elapsed Time Completion Time Compressed Tag
---------- ------------ ------------- --------- ----
DISK        00:01:53     08-SEP-09       YES       TAG20090908T025112

List of Backup Pieces for backup set 58 Copy #2
BP Key   Pc# Status     Piece Name
-------  ---  --------  --------
89       1    AVAILABLE
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnndf_TAG20090908T025112_5bd9ojln_.bkp
BS Key   Type  LV   Size
------   ----  ---  ----
59       Full  1.03M
SPFILE Included:Modification time:08-SEP-09
SPFILE db_unique_name:ROB1
Control File Included: Ckp SCN: 635636      Ckp time:08-SEP-09

Backup Set Copy #1 of backup set 59
Device Type Elapsed Time Completion Time Compressed Tag
---------- ------------ ------------- --------- ----
DISK        00:00:02     08-SEP-09       YES       TAG20090908T025112

List of Backup Pieces for backup set 59 Copy #1
BP Key   Pc# Status     Piece Name
-------  ---  --------  --------
75       1    AVAILABLE
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_ncsnf_TAG20090908T025112_5bd6vpb1_.bkq
Backup Set Copy #2 of backup set 59
Device Type Elapsed Time Completion Time Compressed Tag
---------- ------------ ------------- --------- ----
DISK        00:00:02     08-SEP-09       YES     TAG20090908T025112

List of Backup Pieces for backup set 59 Copy #2
BP Key   Pc# Status     Piece Name
-------  ---  --------  --------
90       1      AVAILABLE
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_ncsnf_TAG20090908T025112_5bd9ozwy_.bkq
```

这个报告提供了所有数据库备份和备份集片标识符的详细列表。可以使用 BS Key 标识符删除整个备份集，也可以使用 BP Key 标识符删除单个备份集片。假设要删除 BS Key ID 为 117 和 118 的两个备份集，只需使用带有 delete 参数的 change 命令来执行这一操作。如果没有使用 FRA，物理备份片将被删除，如果使用 FRA，那么只有当 FRA 中的其他文件被删除时才会删除该物理文件。下面是删除这些备份集的命令示例：

```
RMAN> change backupset 56, 57, 58, 59 delete;
using channel ORA_DISK_1
List of Backup Pieces
BP Key   BS Key  Pc# Cp# Status      Device Type Piece Name
------- ------- --- --- ----------- ----------- ----------
87       56       1   2   AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd9od5x_.bkp
72       56       1   1   AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd6qplq_.bkp
88       57       1   2   AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd9ohdb_.bkp
73       57       1   1   AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd6qydw_.bkp
89       58       1   2   AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnnf_TAG20090908T025112_5bd9ojln_.bkp
74       58       1   1   AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnnf_TAG20090908T025112_5bd6r0mz_.bkp
75       59       1   1   AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_ncsnf_TAG20090908T025112_5bd6vpb1_.bkp
90       59       1   2   AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_ncsnf_TAG20090908T025112_5bd9ozwy_.bkp

Do you really want to delete the above objects (enter YES or NO)? yes
deleted backup piece
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd9od5x_.bkp RECID=16 STAMP=697002060
deleted backup piece
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd6qplq_.bkp RECID=1 STAMP =696999062
deleted backup piece
backup piece
```

```
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd9ohdb_.bkp RECID=17 STAMP=697002063
deleted backup piece
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T025101_5bd6qydw_.bkp  RECID=2 STAMP=696999070
deleted backup piece

backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnnf_TAG20090908T025112_5bd9ojln_.bkp RECID=18 STAMP=677002064
deleted backup piece
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnnf_TAG20090908T025112_5bd6r0mz_.bkp RECID=3 STAMP=696999072
deleted backup piece
backup piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_ncsnf_TAG20090908T025112_5bd6vpb1_.bkp RECID=4 STAMP=696999190
deleted backup piece
backup piece
hande=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_ncsnf_TAG20090908T025112_5bd9ozwy_.bkp RECID=19 STAMP=697002079
deleted backup piece
Deleted 8 objects
```

在这个示例中，备份集 56、57、58、59 以及与这几个备份集关联的所有备份片都会被删除。下面是其他一些删除备份集片的 change 命令的选项示例：

```
change backuppiece 1304 delete;
change archivelog until logseq = 544 delete;
```

最后，可以使用 change backuppiece uncatalog 命令从目录中删除备份集片。如果 change backuppiece uncatalog 命令删除最后剩余的备份集片，它也将删除备份集记录。下面是使用 change backuppiece uncatalog 命令的示例：

```
RMAN> Change backuppiece
'/u01/oracle/RMAN/mydb/mydb_user01_01.bak' uncatalog;
```

RMAN 实习：使用 change 命令

实习注意事项

本实习假定 Oracle 数据库已经运行在 ARCHIVELOG 模式中，此外还假定在磁盘上备份数据库，该数据库中包含一个 USERS 表空间以及一个与 USERS 表空间关联的数据文件。

步骤 1　使用 RMAN 备份 USERS 表空间。

```
RMAN> backup tablespace users;
```

```
Starting backup at 08-SEP-09
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile file number=00004 name=ora01/oracle/rob1/rob1/users01.dbf
channel ORA_DISK_1: starting piece 1 at 08-SEP-09
channel ORA_DISK_1: finished piece 1 at 08-SEP-09
piece
handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnndf_TAG20090908T051140_5bdgzfc9_.bkp tag=TAG20090908T51140
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 08-SEP-09
```

步骤 2 查看备份操作的输出，确定刚刚创建的这个备份集片。在步骤 1 的输出中突出显示了这个备份集片。

步骤 3 使用 list backup of tablespace users 命令确定这个备份集片的备份键，我们需要在控制文件或恢复目录中将这个备份集片标记为 DELETED。注意在下面的输出中，突出显示了备份集键和相关的备份集片。

```
RMAN> list backup of tablespace users;
List of Backup Sets
===================
BS Key  Type LV  Size
------  ---  --- -------
68      Full     1.03M
  List of Datafiles in backup set 68
  File LV Type Ckp SCN     Ckp Time Name
  ---- -- ---- ---------- --------- ----
  4       Full 637442     08-SEP-09 /ora01/oracle/rob1/rob1/users01.dbf

Backup Set Copy #2 of backup set 68
Device Type Elapsed Time Completion Time Compressed Tag
---------- ------------ --------------- ---------- ---
DISK        00:00:01     08-SEP-09       NO         TAG20090908T032815

  List of Backup Pieces for backup set 68 Copy #2
  BP Key   Pc#  Status      Piece Name
  ------  ---- ----------  -----------
  96       1    AVAILABLE
  /oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
  /o1_mf_nnndf_TAG20090908T032815_5bd9pon1_.bkp

Backup Set Copy #1 of backup set 68
Device Type Elapsed Time Completion Time Compressed Tag
---------- ------------ --------------- ---------- ---
DISK        00:00:01     08-SEP-09       NO         TAG20090908T032815

  List of Backup Pieces for backup set 68 Copy #1
```

```
  BP Key   Pc#  Status       Piece Name
  ------  ---- ----------   -----------
  84         1    AVAILABLE
  /oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
  /o1_mf_nnndf_TAG20090908T032815_5bd8xhxm_.bkp

BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
176     Full     5.07M       DISK        00:00:01     08-SEP-09
BP Key: 179 Status: AVAILABLE Compressed: NO Tag: TAG20090908T040309
Piece Name: /oracle/backup/0hkomrbt_1_1
  List of Datafiles in backup set 176
  File LV Type Ckp SCN     Ckp Time Name
  ---- -- ---- ---------- --------- ----
  4       Full 642222     08-SEP-09  /ora01/oracle/rob1/rob1/user01.dbf

BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
421     Full     5.52M       DISK        00:00:00     08-SEP-09
BP Key: 424 Status: AVAILABLE Compressed: NO Tag: TAG20090908T051140
  Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnndf_TAG20090908T051140_5bdgzfc9_.bkp
  List of Datafiles in backup set 421
  File LV Type Ckp SCN     Ckp Time Name
  ---- -- ---- ---------- --------- ----
  4       Full 644956     08-SEP-09  /ora01/oracle/rob1/rob1/user01.dbf
```

步骤 4 退回到操作系统并在备份集片上建立一个目录清单。我们应该可以看到该备份集片。

步骤 5 使用 change backuppiece 命令将这个备份集片的状态标志由 AVAILABLE 变为 DELETED。

```
RMAN> change backuppiece 96 delete;
using channel ORA_DISK_1
List of Backup Pieces
BP Key   BS Key  Pc# Cp# Status      Device Type Piece Name
------- ------- --- --- ----------- ----------- ----------
96         68    1    2  AVAILABLE   DISK
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnndf_TAG20090908T032815_5bd9pon1_.bkp
Do you really want to delete the above objects(enter YES or NO)? yes
deleted backup piece
backup piece
  handle=/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnndf_TAG20090908T032815_5bd9pon1_.bkp  RECID=25 STAMP=697002101
Deleted 1 objects
```

步骤 6 再次使用 list backup of tablespace users 命令，确定在恢复操作期间不能再使

用这个备份集片。

```
RMAN> list backup of tablespace users;
```

步骤 7 退回到操作系统并在备份集片上建立一个目录清单。我们应该不会再看到该备份集片。Oracle 会删除已有的备份集片。

16.1.5 delete 命令

备份集不是永远存在的。我们可以使用保留策略标记备份有效性和生命周期的结束。正如前面所介绍的，冗余策略的实施不会从 RMAN 目录中删除备份，而只是将这些备份标记为过时(OBSOLETE)状态，这与本章前面讨论的 crosscheck 命令完全相同。crosscheck 命令可以将过时的备份和副本标记为 EXPIRED 状态，但是不会删除这些备份和副本。

delete 命令对备份和副本的影响很大。通过执行 delete 命令，可以删除基于保留标准被标记为过时的任何备份，还可以将恢复目录或控制文件中的备份从 EXPIRED 状态变为 DELETED 状态。下面是一组 delete 命令的使用示例：

```
delete expired;
delete obsolete;
```

当执行 delete 命令时，关联的 RMAN 备份记录将从恢复目录中删除。可以通过观察恢复目录的视图 WR_BACKUP_PIECE 知道。在数据库控制文件中包含的这些记录将被保留在控制文件中直到被重写为止。这些记录将带有 D 状态出现在 V$BACKUP_PIECE 视图中。这些删除记录将不会出现在 RMAN 命令输出中，例如 list backup of database summary 命令中。

执行 delete 命令时，RMAN 会请求用户确认这个删除指令。如果确认了这个删除指令，RMAN 就会完成 delete 操作。可以使用 delete 命令的 noprompt 选项避免验证执行 delete 命令的需要。如果写出 RMAN 工作的脚本这就变的方便了。下面是使用带有 noprompt 选项的 delete 命令的示例：

```
delete noprompt obsolete;
```

注意：
如果一个备份被标记为 DELETED 状态，就不能恢复这个备份。如果该备份物理可用，我们仍然可以使用 catalog 命令在控制文件/恢复目录上注册备份集来恢复这个备份。

RMAN 实习：delete 命令的使用

实习注意事项

这个实习建立在前一个实习(“change 命令的使用”)的基础上，使用 crosscheck 命令进行处理。

步骤 1　确定备份集片丢失后，要将这个备份集片的状态标记为永久丢失。在 RMAN 提示符下执行 delete expired backup 命令。

```
RMAN> delete expired backup;
using channel ORA_DISK_1
using channel ORA_DISK_2
List of Backup Pieces
BP Key  BS Key   Pc# Cp# Status       Device Type Piece Name
------- ------- --- --- ----------- ----------- ----------
53      53       1   1   EXPIRED      DISK
D:\BACKUP\RECOVER\BACKUP_1VE25VC0_1_1
Do you really want to delete the above objects (enter YES or NO)?
```

步骤 2　在报告中查看将被标记为 DELETED 状态的对象。如果这些对象都能被标记为 DELETED 状态，就可以在 RMAN 提示符下回答 YES 并按 Enter 键。在输出中查看是否成功地执行了 delete 操作。

```
deleted backup piece
backup piece handle=D:\BACKUP\RECOVER\BACKUP_1VE25VC0_1_1 recid=53
stamp=472055171
Deleted 1 EXPIRED objects
```

16.1.6　在 RMAN 中为其他备份编写目录

使用 catalog 命令可以在 RMAN 中记录数据文件备份、归档日志备份和控制文件备份，随后还能使用这些备份来还原和恢复数据库。Oracle Database 也允许使用 catalog 命令来为控制文件中已有的备份集片编写目录。当使用一个不包含最新的 RMAN 信息的较早备份控制文件还原数据库时，则可以很方便地使用该功能来完成操作。

下面是一些使用 catalog 命令为旧数据文件备份编写目录的示例：

```
-- first, backup the users tablespace
sqlplus sys/robert as sysdba
alter tablespace users begin backup;
host copy d:\oracle\oradata\recover\users01.dbf
d:\backup\recover\users01.dbf.backup
alter tablespace users end backup;
alter system archive log current;
host copy d:\oracle\admin\recover\arch\*.* d:\backup\recover
-- get a list of archivelog files that were created
host dir d:\backup\recover
alter database backup control file to 'd:\backup\recover.ctl'
quit

-- Now, catalog the backup in rman
rman target sys/robert
catalog datafilecopy 'd:\backup\recover\users01.dbf.backup';
-- Replace arc001.log with the list of archive logs you generated earlier
```

```
catalog archivelog 'd:\backup\recover\arc001.log';
-- Now catalog the control file.
catalog controlfilecopy 'd:\backup\recover.ctl';
```

catalog 命令允许将与新的备份集相关的信息输入到控制文件或恢复目录中。RMAN 将重写与编写目录的信息冲突的已存在的目录信息。如果需要移动备份集片的位置，该命令就非常方便。在该示例中，我们已经将所有的备份集片移动到新的目录。可以使用 catalog 命令来为控制文件中的每个移动的备份集片加载正确的目录位置。

```
RMAN> catalog backuppiece '/opt/oracle/oracle-10.0.0/dbs/backup';
```

也可以使用带有 start with 选项的 catalog 命令，该命令允许定义包含将要编写目录的 RMAN 备份集片的目录。RMAN 然后对该目录中的所有备份集片编写目录。下面是以这种方式使用 catalog 命令的一个示例：

```
RMAN> catalog start with '/u01/oracle/RMAN/mydb';
```

一旦按下 Enter 键，该命令就会提示编写目录的文件列表，并询问是否希望对所列出的文件编写目录。如果确定编写目录，RMAN 就会对列出的所有备份集片编写目录(这些备份集片包含在/u01/oracle/RMAN/mydb 目录中)。该命令也允许对一些具有类似名称的备份集片编写目录。例如，如果需要对按备份顺序命名(例如 backupset01、backupset02 等)的一些备份集片编写目录，则可以执行如下命令：

```
RMAN> catalog start with '/u01/oracle/RMAN/mydb/backup';
```

当使用 catalog start with 命令时，不会局限于对特定的文件编写目录；该命令将尝试对符合参数列表的所有文件编写目录。然而，在执行 catalog 进程的过程中，非备份集片的文件将会使编写目录的进程失败，并且产生一个错误。作为备份集片的文件将被成功地编写目录，而不管是否会产生其他错误。

16.2 RMAN 存储的脚本

如果经常需要执行相同的 RMAN 操作，您可能希望存储这些操作，并在需要时执行这些存储的操作。当然，我们可以创建一个命令文件，这是一个在磁盘上物理存在的文本文件，包含 RMAN 命令。如下所示，可以使用 cmdfile 参数从 RMAN 命令行接口执行该命令文件：

```
rman target robert/password cmdfile=run_backup.cmd
```

此外，还可以使用@命令在 RMAN 中运行该命令文件：

```
@@run_backup.cmd
```

RMAN 提供了另一个选项，即在恢复目录中存储脚本，这个操作要求使用恢复目录。因此，如果没有使用恢复目录，就不能存储 RMAN 脚本。本节将介绍在恢复目录中存储

脚本以及管理这些脚本的方法。

16.2.1　创建存储的脚本

使用 create script RMAN 命令可以在恢复目录中存储脚本。创建每个存储的脚本时，都要为脚本指定一个名称。可以创建执行数据库备份、恢复和维护操作的脚本。要创建脚本，就必须连接到恢复目录。这里是使用 create script 命令创建备份脚本的示例。RMAN 也允许使用 comment 参数存储与存储脚本相关的注释：

```
create script my_backup_script
    comment 'This script backs up the database'
    { backup database plus archivelog;}
```

Oracle Database 11*g* 支持替代变量的使用。每个替代变量定义为一个&(and)符号和一个数字，使得每个变量是唯一的。例如，可以重写下面的脚本：

```
creat script my_backup_script
    comment 'This script backs up the database'
    {backup database tag '&1' plus archivelog;}
```

当执行这个命令时，RMAN 将提示用户替代变量的初始值。

16.2.2　查询存储脚本信息的恢复目录

可以查询 RC_STORDE_SCRIPT 视图来使用恢复目录视图确定脚本的名称，这些脚本的名称储存在恢复目录中。可以查询 RC_STORED_SCRIPT_LINE 视图来查看给定脚本的内容。

16.2.3　修改存储的脚本

使用 replace script 命令可以替换恢复目录中存储的脚本。下面是一个使用 replace script 命令的示例。需要注意的是，我们也向该脚本添加了一条注释：

```
replace script my_backup_script
    comment 'This script backs up the database'
    { backup database plus archivelog delete input;}
```

16.2.4　删除存储的脚本

使用 delete script 命令可以删除一个存储的脚本。我们必须连接到恢复目录，才能成功地删除存储的脚本。下面是一个使用 delete script 命令的示例：

```
delete script my_backup_script;
```

16.2.5　使用存储的脚本

创建了一些存储的脚本后，可以执行 execute script 命令来使用这些脚本。如下所示，应该连接到恢复目录，并且在 run 代码块的范围内使用 execute script 命令。

```
run {execute script my_backup_script;}
```

如果使用替代变量，就可以使用 using 参数(在 execute script 命令里包含这些参数的值)。如下所示：

```
Run {execute script my_backup_script using TEST_BACKUP;}
```

16.2.6 打印存储的脚本

如果要打印一个存储的脚本的副本，则可以使用 print script 命令。如下所示，可以连接到恢复目录并运行 print script 命令：

```
RMAN> print script my_backup_script;

printing stored script: my_backup_script
{ backup database plus archivelog;}
```

还可以使用 RC_STORED_SCRIPT_LINE 恢复目录视图来显示存储的脚本的内容，如下所示：

```
select script_name, text from rc_stored_script_line
order by script_name, line;
SCRIPT_NAME       TEXT
---------------- ----------------------------------------
my_backup_script  { backup database plus archivelog;}
```

RMAN 实习：使用 RMAN 存储的脚本

实习注意事项

本实习需要运行一个名为 recover 的 Oracle 数据库，还需要使用一个名为 catalog 的 Oracle 数据库来存储恢复目录。

步骤 1 连接到目标数据库和恢复目录。

```
rman target rman_account/rman_password catalog
rcat_user/rcat_password@catalog
```

步骤 2 创建一个备份目标数据库的存储的脚本。

```
RMAN> create script my_backup_script
2> {backup database plus archivelog;}
created script my_backup_script
```

步骤 3 打印这个存储的脚本。

```
RMAN> print script my_backup_script;
printing stored script: my_backup_script
{backup database plus archivelog;}
```

步骤 4　运行这个存储的脚本，备份目标数据库。

```
RMAN> run {execute script my_backup_script;}
```

步骤 5　删除这个存储的脚本。

```
RMAN>delete script my_backup_script;
```

16.3　删除数据库

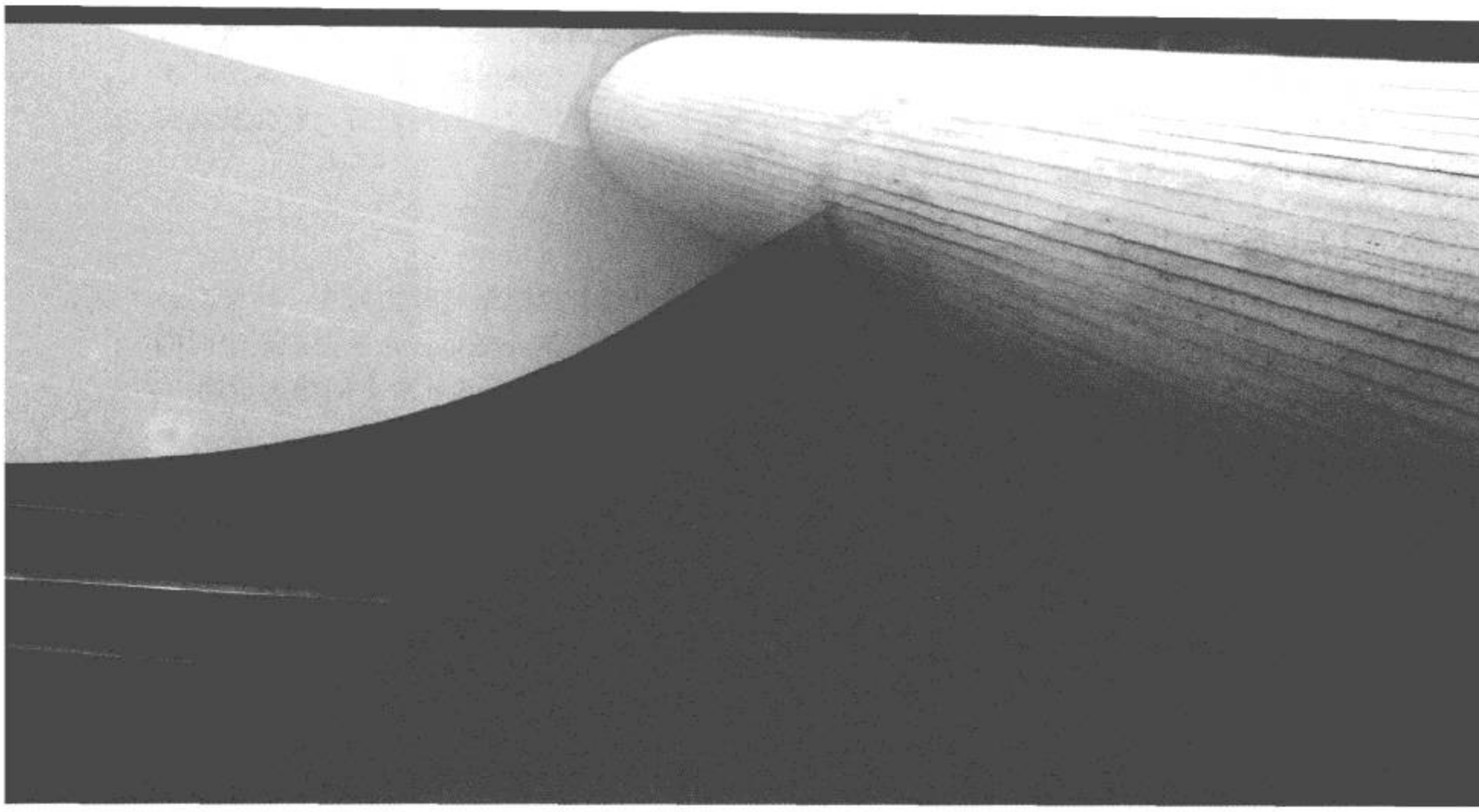
如果无法使用其他方法来处理数据库的问题，则可以使用 RMAN 提供的命令 drop database。只需要将数据库置于受限制会话模式，使用 RMAN 连接到目标数据库，执行 drop database 命令，就可以简单地删除数据库。我们可以添加 including backups 参数，则会同时删除 RMAN 相关的备份。执行该命令时，RMAN 将首先确认操作，然后继续删除数据库。如果希望不显示提示，则可以使用 noprompt 参数。下面是使用 drop database 命令的示例：

```
SQL> alter system enable restricted session;
SQL> quit;
.. log into RMAN ..
RMAN>drop database including backups;
```

16.4　小结

本章首先讲述了维护 RMAN 正常运行的各种操作。我们讨论了两个非常重要的操作：使用 crosscheck 命令和验证 RMAN 备份。我们也讨论了保留策略以及 RMAN 如何使用这些保留策略来控制备份的保留时间。

我们还介绍了 change 和 delete 命令以及如何用它们来修改控制文件或恢复目录中 RMAN 记录的状态，然后介绍了向控制文件或恢复目录添加备份。最后，我们介绍了恢复目录的维护以及将存储的脚本用于 RMAN 的各种操作。



第 17 章

RMAN 的监控和报告

由于用户需要确切知道数据库已备份并且当前可恢复，所以 RMAN 提供了一些优秀的报告工具。本章详细介绍了 RMAN 报告。首先，将介绍 RMAN 的 list 和 report 命令，这些命令提供了深入分析使用 RMAN 备份的数据库以及数据库备份的适用工具，并且是从 RMAN 中提取信息的主要方法。您会发现这些列表和报告不仅为恢复操作提供方便，而且还能方便地查看 RMAN 的配置和其他管理任务(如判断是否备份了表空间)。

17.1 RMAN 的 list 命令

RMAN 的 list 命令是一种在数据库控制文件或恢复目录中查询备份的历史信息的方

法。列表提供了一组信息，包括从数据库对应物的列表到备份集和归档日志备份的列表的所有信息。一般情况下，只有我们想要知道数据库是否已备份以及什么时候备份时才会生成一个列表。列表格式最初看上去没有头绪，不过查看完一些列表后，您会发现实际上很容易理解。下面，我们将介绍 list 命令及其作用。

17.1.1 列出对应物

list incarnation 命令为用户提供了目标数据库的每个数据库对应物列表，如果有必要，可以使用这个列表的信息将数据库恢复到执行最后一条 resetlogs 命令前的某个时间点(参阅第 14 章有关该操作的详细介绍)。下面是一个 list incarnation 命令的输出示例：

```
RMAN> list incarnation of database;
using target database control file instead of recovery catalog
List of Database Incarnations
DB Key  Inc Key  DB Name  DB ID            STATUS  Reset SCN  Reset Time
------- ------- -------- ---------------- ------- ---------- ----------
1       1       ROB1     1854903786       PARENT  1          07-SEP-09
2       2       ROB1     1854903786       CURRENT 635384     08-SEP-09
```

在这个清单中，可以看到数据库具有 2 个不同的对应物，报告中的每条记录都代表一个对应物。每个对应物都具有自己的键(Inc Key)，如果要重置数据库对应物，就会用到该键(参阅第 14 章)。从这个报告中还可以获取数据库名称和 ID。

STATUS 列列出的是对应物的状态，该状态表明对应物是旧的对应物(PARENT)、当前的对应物(CURRENT)，还是孤立的对应物(ORPHAN，即在 resetlogs 命令之后进行恢复)。最后，Reset SCN 和 Reset Time 列主要说明创建数据库对应物的时间(这是第一个条目的 Reset SCN 为 1 的原因)。该列对使用 resetlogs 命令进行恢复有用，并使得能较容易地恢复到前面的对应物。

特别需要注意的一点是，使用恢复目录和不使用恢复目录分别生成的输出通常存在差异。例如，下面是一个连接恢复目录而执行 list incarnation 命令的输出：

```
RMAN> list incarnation of database;
  List of Database Incarnations
DB Key  Inc Key  DB Name  DB ID            STATUS   Reset SCN  Reset Time
------- ------- -------- ---------------- ------- ---------- ----------
2       18      ROB1     1854903786       PARENT   1          07-SEP-09
2       4       ROB1     1854903786       CURRENT  635384     08-SEP-09
```

在该示例中，DB Key 和 Inc Key 与使用控制文件所生成的报告是不同的，因此许多报告具有能够标识报告中特定项的键。可以在其他 RMAN 命令(如 reset database 命令)中使用这些键。由于这些键值会根据是否连接到恢复目录而变化，所以必须注意需要的是哪些键。

17.1.2 列出备份

list 命令提供了许多不同的选项来报告数据库备份和副本的状态。本节将介绍几个这

样的报告。

1. 概述可用的备份

首先，将介绍获取备份概述信息的方法。list 命令提供了两个选项。第一个选项是 list backup summary 命令，如下所示：

```
RMAN> list backup summary;
List of Backups
===============
Key     TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
------- -- -- - ----------- --------------- ------- ------- ---------- ---
60      B  A  A DISK        08-SEP-09       1       2       YES TAG20090908T025311
61      B  A  A DISK        08-SEP-09       1       2       NO  TAG20090908T025326
62      B  F  A DISK        08-SEP-09       1       2       NO  TAG20090908T025328
63      B  A  A DISK        08-SEP-09       1       2       NO  TAG20090908T025355
67      B  A  A DISK        08-SEP-09       1       2       NO  TAG20090908T032531
68      B  F  A DISK        08-SEP-09       1       1       NO  TAG20090908T032815
70      B  F  A DISK        08-SEP-09       1       1       NO  TAG20090908T032850
176     B  F  A DISK        08-SEP-09       1       1       NO  TAG20090908T040309
207     B  F  A DISK        08-SEP-09       1       1       NO  TAG20090908T040315
421     B  F  A DISK        08-SEP-09       1       1       NO  TAG20090908T051140
433     B  F  A DISK        08-SEP-09       1       1       NO  TAG20090908T051144
```

上面的报告向用户提供了一些有用的概述信息。Key 列出了备份集键；TY(类型)和 LV(级别)列指示列出的备份的类型(B 表示备份[backup]，F 表示完整备份[full]，A 表示归档日志[archive log]，0 和 1 表示增量备份[incremental backup])；S 列说明备份的状态(可用，AVAILABLE；不可用，UNAVAILABLE；到期，EXPIRED)；Device Type 列说明备份是磁带备份还是磁盘备份。此外，我们还能查看备份生成的日期(Completion Time 列)，组成备份集的备份片数目(#Pieces 列)或副本数目(#Copies 列)，以及为这个备份集分配的标记(Tag 列)。

大多数 list 命令都可以在末尾使用 summary 参数，如下所示：

```
list backup of database summary;
list expired backup of archivelog all summary;
list backup of tablespace users summary;
```

2. 按数据库文件类型列出备份

概述备份的另一种方法是使用 list backup by file 命令列出每个备份集和备份集片。下面是一个执行该命令的报告示例(为简略起见，删除了一些输出)：

```
RMAN> list backup by file;
List of Datafile Backups
========================
File  Key  TY LV S Ckp SCN  Ckp Time #Pieces #Copies Compressed Tag
----- ---- - -- - -------- ------- ------- ------- ---------- ---
2     62   B  F  A 635676   08-SEP-09 1     2       NO  TAG20090908T025328
```

```
4     421   B  F  A 644956   08-SEP-09 1       1       NO TAG20090908T051140
      176   B  F  A 642222   08-SEP-09 1       1       NO TAG20090908T040309
      68    B  F  A 637442   08-SEP-09 1       1       NO TAG20090908T032815

List of Archived Log Backups
============================
Thrd Seq   Low SCN   Low Time  BS Key S #Pieces #Copies Compressed Tag
---- ----- -------- -------- ------- - ------- ------- ---------- ---
1    2     635574   08-SEP-09 60      A 1       2       YES TAG20090908T025311
1    3     635642   08-SEP-09 61      A 1       2       NO  TAG20090908T025326
1    4     635668   08-SEP-09 63      A 1       2       NO  TAG20090908T025355
1    7     636872   08-SEP-09 67      A 1       2       NO  TAG20090908T032531
1    8     637317   08-SEP-09 67      A 1       2       NO  TAG20090908T032531
1    9     637320   08-SEP-09 67      A 1       2       NO  TAG20090908T032531
1    10    637324   08-SEP-09 67      A 1       2       NO  TAG20090908T032531

List of Control File Backups
============================
CF Ckp SCN Ckp Time   BS Key  S #Pieces #Copies Compressed Tag
---------- --------- ------- - ------- ------- ---------- ---
644990     08-SEP-09 433     A 1       1       NO   TAG20090908T051144
642268     08-SEP-09 207     A 1       1       NO   TAG20090908T040315
637490     08-SEP-09 70      A 1       1       NO   TAG20090908T032850

List of SPFILE Backups
======================
Modification Time BS Key  S #Pieces #Copies Compressed Tag
----------------- ------- - ------- ------- ---------- ---
08-SEP-09         433     A 1       1       NO         TAG20090908T051144
08-SEP-09         207     A 1       1       NO         TAG20090908T040315
08-SEP-09         70      A 1       1       NO         TAG20090908T032850
```

上面的报告按照备份类型(数据文件备份、归档日志备份、控制文件备份以及服务器参数文件备份)概述了每个备份文件，然后按数据文件备份。在这个报告中，我们可以获取备份日期和与备份文件关联的特定键。根据备份类型，可以得到属于这个备份类型的信息。

3. 其他备份信息

如果想得到 RMAN 备份的更多信息，就需要使用 list backup 命令。list backup 命令提供所生成备份(包括备份集、归档的重做日志备份、控制文件备份以及服务器参数文件备份)的详细信息。下面将介绍一个执行 list backup 命令的报告示例：

```
RMAN> list backup;
BS Key Size       Device Type Elapsed Time Completion Time
------ ---------- ----------- ------------ ---------------
509    12.43M     DISK        00:00:04     08-SEP-09
```

```
        BP Key: 513 Status: AVAILABLE Compressed: YES Tag: TAG20090908T192844
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T192844_5bg16fk5_.bkp

  List of Archived Logs in backup set 509
  Thrd Seq     Low SCN    Low Time  Next SCN   Next Time
  ---- ------- ---------- --------- ---------- ---------
  1    7       636872     08-SEP-09 637317     08-SEP-09
  1    8       637317     08-SEP-09 637320     08-SEP-09
  1    10      637324     08-SEP-09 637349     08-SEP-09
  1    11      637349     08-SEP-09 675660     08-SEP-09
  1    12      675660     08-SEP-09 676291     08-SEP-09
```

该清单是归档日志备份，其备份集键(BS Key)为 509。该清单中列出了该备份的大小，并且表明备份到磁盘而不是 SBT。备份使用的时间非常短，只有 4 秒，并且备份完成时间是 9 月 8 日。在该报告的后面表明该备份可用，并且是一个压缩的备份。我们也可以看到备份集片名，备份集片名提供了该备份的物理位置。在该报告的最后显示了归档的重做日志的列表，这些是包含在该备份集中的归档的重做日志。下面是该备份的剩余报告的示例：

```
BS Key Type LV Size       Device Type Elapsed Time Completion Time
------ ---- -- ---------- ----------- ------------ ---------------
510    Full    243.36M    DISK        00:01:16     08-SEP-09
       BP Key: 514 Status: AVAILABLE Compressed: YES Tag: TAG20090908T192853
       Piece Name:/oracle/app/oracle/flash_recovery_area/ROB1/backupset/
       2009_09_08/o1_mf_nnndf_TAG20090908T192853_5bg16pwv_.bkp

  List of Datafiles in backup set 510
  File LV Type Ckp SCN  Ckp Time  Name
  ---- -- ---- -------- --------- ----
  1       Full 676334   08-SEP-09 /ora01/oracle/rob1/rob1/system01.dbf
  2       Full 676334   08-SEP-09 /ora01/oracle/rob1/rob1/sysaux01.dbf
  3       Full 676334   08-SEP-09 /ora01/oracle/rob1/rob1/undotbs01.dbf
  4       Full 676334   08-SEP-09 /ora01/oracle/rob1/rob1/users01.dbf

BS Key Size       Device Type Elapsed Time Completion Time
------ ---------- ----------- ------------ ---------------
536    783.00K    DISK        00:00:01     08-SEP-09
       BP Key: 541 Status: AVAILABLE Compressed: YES Tag: TAG20090908T193014
       Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T193014_5bg196xr_.bkp

List of Archived Logs in backup set 536
Thrd Seq     Low SCN    Low Time  Next SCN   Next Time
---- ------- ---------- --------- ---------- ---------
1    13      676291     08-SEP-09 676364     08-SEP-09
```

```
BS Key Type LV Size       Device Type Elapsed Time Completion Time
------ ---- -- ---------- ----------- ------------ ---------------
555    Full    9.67M      DISK        00:00:01     08-SEP-09
       BP Key: 557 Status: AVAILABLE Compressed: NO Tag: TAG20090908T193017
       Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697059017_5bg19b6p_.bkp
  SPFILE Included: Modification time: 08-SEP-09
  SPFILE db_unique_name: ROB1
  Control File Included: Ckp SCN: 676443       Ckp time: 08-SEP-09
```

这是实际的数据库备份，其中的输出类似于前面的输出，但是现在获得包含在该备份中的所有数据文件的完整列表。我们可以看到，数据文件备份实际由一个备份集片(BS Key 为 510)组成。进行恢复时，RMAN 只关心需要恢复的最近的时间点。一旦知道了该时间点，RMAN 就会选择最佳的备份并用于执行恢复。这是非常重要的信息。

在该列表中，也可以找到归档日志备份(BS Key 为 536)，该归档日志备份中仅包含一个归档日志。在该报告的结尾部分，可以看到控制文件/服务器参数文件(BS Key 为 555)的自动备份。通过报告输出中的“SPFILE Included”和“Control File Include”语句可以知道该备份是一个自动备份。通过以下示例可更深入地了解归档日志备份的输出：

```
BS Key  Size       Device Type Elapsed Time Completion Time
------ ---------- ----------- ------------ ---------------
536    783.00K    DISK        00:00:01     08-SEP-09
       BP Key: 541 Status: AVAILABLE Compressed: YES Tag: TAG20090908T193014
       Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T193014_5bg196xr_.bkp

  List of Archived Logs in backup set 536
  Thrd Seq     Low SCN    Low Time  Next SCN   Next Time
  ---- ------- ---------- --------- ---------- ---------
  1    13      676291     08-SEP-09 676364     08-SEP-09
```

上例所示的备份集的备份集键为 536。其头信息与前述的备份集非常类似。然而，该备份集是归档日志备份，在随后列出的行中，RMAN 提供了在备份集中备份的归档重做日志列表。列出了归档重做日志的线程和顺序号，LOW SCN 和时间以及 NEXT SCN 和时间。利用 LOW TIME/SCN 和 HIGH(或者报告中列出的 NEXT SCN)TIME/SCN 及相关的时间信息可以确定归档日志建立的时间。

接下来查看一个增量备份集报告：

```
BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
857     Incr 1   1000.00K   DISK        00:00:27     08-SEP-09
        BP Key: 861 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202840
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
```

```
/o1_mf_nnnd1_TAG20090908T202840_5bg4psxw_.bkp
  List of Datafiles in backup set 857
  File LV Type Ckp SCN    Ckp Time Name
  ---- -- ---- --------- --------- ----
  1     1  Incr 679212    08-SEP-09 /ora01/oracle/rob1/rob1/system01.dbf
  2     1  Incr 679212    08-SEP-09 /ora01/oracle/rob1/rob1/sysaux01.dbf
  3     1  Incr 679212    08-SEP-09 /ora01/oracle/rob1/rob1/undotbs01.dbf
  4     1  Incr 679212    08-SEP-09 /ora01/oracle/rob1/rob1/users01.dbf
```

这个报告非常类似于其他的报告，它们之间唯一的差别在于最后一个报告片段中，Type 字段中的 Incr 用来代表增量备份，同时 LV(级别 level)列说明这个增量备份的级别。如果增量备份的级别为 0，LV 列就会显示与级别为 0 的基本备份相对应的数字 0。

4. 列出适合恢复的备份

如果需要查看能够用于还原和恢复数据库的所有数据文件备份或副本，则可以使用 list recoverable 命令。list 命令提供了状态为 AVAILABLE 的、可用于还原数据库的所有备份列表(该列表只用于当前的对应物)，其中包括了备份、映像副本和增量备份。如果增量备份没有有效的父备份，它就不会包括在备份中。

```
RMAN> list recoverable backup of database;
    List of Backup Sets
    ===================
    BS Key  Type LV Size       Device Type Elapsed Time Completion Time
    ------- ---- -- ---------- ----------- ------------ ---------------
    713     Incr 0 243.27M DISK 00:01:10 08-SEP-09
            BP Key: 715 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202601
            Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnnd0_TAG20090908T202601_5bg4ktk1_.bkp

      List of Datafiles in backup set 713
      File LV Type Ckp SCN   Ckp Time Name
      ---- -- ---- -------- --------- ----
      1     0  Incr 678900    08-SEP-09 /ora01/oracle/rob1/rob1/system01.dbf
      2     0  Incr 678900    08-SEP-09 /ora01/oracle/rob1/rob1/sysaux01.dbf
      3     0  Incr 678900    08-SEP-09 /ora01/oracle/rob1/rob1/undotbs01.dbf
      4     0  Incr 678900    08-SEP-09 /ora01/oracle/rob1/rob1/users01.dbf

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
857     Incr 1  1000.00K    DISK        00:00:27     08-SEP-09
        BP Key: 861 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202840
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnnd1_TAG20090908T202840_5bg4psxw_.bkp

  List of Datafiles in backup set 857
```

```
File LV Type Ckp SCN   Ckp Time Name
---- -- ---- -------- --------- ----
1    1  Incr 679212   08-SEP-09 /ora01/oracle/rob1/rob1/system01.dbf
2    1  Incr 679212   08-SEP-09 /ora01/oracle/rob1/rob1/sysaux01.dbf
3    1  Incr 679212   08-SEP-09 /ora01/oracle/rob1/rob1/undotbs01.dbf
4    1  Incr 679212   08-SEP-09 /ora01/oracle/rob1/rob1/users01.dbf
```

5. 列出到期的备份

执行 list backup 命令会显示可用的和到期的备份集。如果只想列出到期的备份，则可以使用 expired 关键字，如下所示：

```
RMAN> list expired backup;


List of Backup Sets
===================
BS Key  Size        Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ---------------
1025    489.00K     DISK        00:00:01     08-SEP-09
        BP Key: 1028 Status: EXPIRED Compressed: NO Tag: TAG20090908T203418
        Piece Name: /oracle/app/oracle/flash_recovery_area/ROB1/backupset/
        2009_09_08/o1_mf_annnn_TAG20090908T203418_5bg51c2c_.bkp
List of Archived Logs in backup set 1025
  Thrd Seq     Low SCN    Low Time  Next SCN   Next Time
  ---- ------- ---------- --------- ---------- ---------
  1    18      679230     08-SEP-09 679716     08-SEP-09
```

该报告提供了所有到期的备份集类型。使用 list expired backup 命令也可以获得到期的表空间和数据文件备份列表，如果插入正确的关键字(例如 list expired backup of datafile 3 或 list expired backup of archivelog all)，则可以获得到期的归档日志备份和控制文件/服务器参数文件自动备份列表。

6. 按表空间名和数据文件号列出备份

执行 list backup of tablespace 或 list backup of datafile 命令的输出与执行 list backup 命令的输出非常相似，这两种 list backup 命令可以列出指定表空间或数据文件的输出，如下所示：

```
RMAN> list backup of tablespace users;
    List of Backup Sets
    ===================
    BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
    ------- ---- -- ---------- ----------- ------------ ---------------
    713     Incr 0   243.27M    DISK        00:01:10     08-SEP-09
            BP Key: 715 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202601
            Piece Name:
    oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
```

```
/o1_mf_nnnd0_TAG20090908T202601_5bg4ktk1_.bkp

  List of Datafiles in backup set 713
  File LV Type Ckp SCN    Ckp Time Name
  ---- -- ---- ---------- --------- ----
  4    0  Incr 678900     08-SEP-09 /ora01/oracle/rob1/rob1/users01.dbf
BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
857     Incr 1  1000.00K   DISK        00:00:27     08-SEP-09
        BP Key: 861 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202840
        Piece Name:
        /oracle/app/oracle/flash_recovery_area/ROB1/backupset/
        2009_09_08/o1_mf_nnnd1_TAG20090908T202840_5bg4psxw_.bkp
  List of Datafiles in backup set 857
  File LV Type Ckp SCN    Ckp Time Name
  ---- -- ---- ---------- --------- ----
  4    1  Incr 679212     08-SEP-09 /ora01/oracle/rob1/rob1/users01.dbf
```

在该示例中需要注意的是，这个特定的备份已经到期，它可能是我们特别关注的备份，特别是在 USERS 表空间中只显示该备份的情况下！我们可以使用 expired 关键字来列出到期的备份(执行 list expired backup of tablespace 命令)。

同样地，执行 list backup of datafile 命令能够列出指定的数据文件的备份：

```
RMAN> list backup of datafile 3;
List of Backup Sets
===================

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
713     Incr 0  243.27M     DISK        00:01:10     08-SEP-09
        BP Key: 715 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202601
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnnd0_TAG20090908T202601_5bg4ktk1_.bkp

  List of Datafiles in backup set 713
  File LV Type Ckp SCN   Ckp Time Name
  ---- -- ---- -------- --------- ----
  3    0  Incr 678900    08-SEP-09 /ora01/oracle/rob1/rob1/undotbs01.dbf

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
857     Incr 1  1000.00K    DISK        00:00:27     08-SEP-09
        BP Key: 861 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202840
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_nnnd1_TAG20090908T202840_5bg4psxw_.bk

  List of Datafiles in backup set 857
```

```
File LV Type Ckp SCN   Ckp Time Name
---- -- ---- -------- --------- ----
3    1  Incr 679212    08-SEP-09 /ora01/oracle/rob1/rob1/undotbs01.dbf
```

如果试图按照时间点还原，并且得到的错误指出无法发现备份或者副本时，使用 list 命令将会带来好处。在此情况下，试着使用带有相同的 until 子句的 list 命令查看是否列出任何可用的备份，通过 until 子句可以发现可能存在的问题，调整 until 子句可以确定哪些为真正可用的时间点恢复。

7. 列出归档日志备份

RMAN 提供了列出归档日志备份的几种选项。要得到磁盘上当前存在的归档日志(这并不意味着它们已被备份)的完整概述，可以使用如下所示的 list archivelog all 命令：

```
RMAN> list archivelog all;
List of Archived Log Copies for database with db_unique_name ROB1
=====================================================================

Key     Thrd Seq     S Low Time
------- ---- ------- - ---------
54      1    9       X 08-SEP-09
        Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_9_5bd8qv45_.arc

1170    1    19      A 08-SEP-09
        Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_19_5bg61141_.arc
```

上面的报告给出了 Oracle 创建的等待被备份的每个归档的重做日志的列表，并且提供了这些归档的重做日志的线程号和序列号。

如果要得到更详细的报告，可以执行 list backup of archivelog all 命令，如下所示：

```
RMAN> list backup of archivelog all;

List of Backup Sets
===================
BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ---------------
712     1.53M      DISK        00:00:00     08-SEP-09
        BP Key: 714 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202600
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T202600_5bg4kr90_.bkp

  List of Archived Logs in backup set 712
  Thrd Seq     Low SCN    Low Time  Next SCN   Next Time
```

```
  ---- ------- ---------- --------- ---------- ---------
  1    14      676364     08-SEP-09 678853     08-SEP-09

BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ---------------
744     61.50K     DISK        00:00:00     08-SEP-09
        BP Key: 749 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202721
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T202721_5bg4n9qy_.bkp

  List of Archived Logs in backup set 744
  Thrd Seq     Low SCN    Low Time  Next SCN    Next Time
  ---- ------- ---------- --------- ---------- ---------
  1    15      678853     08-SEP-09 678935     08-SEP-09

BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ---------------
856     370.50K    DISK        00:00:00     08-SEP-09
        BP Key: 860 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202838
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T202838_5bg4pq6x_.bkp

  List of Archived Logs in backup set 856
  Thrd Seq     Low SCN    Low Time  Next SCN    Next Time
  ---- ------- ---------- --------- ---------- ---------
  1    16      678935     08-SEP-09 679163     08-SEP-09

BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ---------------
894     63.00K     DISK        00:00:01     08-SEP-09
        BP Key: 899 Status: AVAILABLE Compressed: YES Tag: TAG20090908T202920
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T202920_5bg4r0y0_.bkp

  List of Archived Logs in backup set 894
  Thrd Seq     Low SCN    Low Time  Next SCN    Next Time
  ---- ------- ---------- --------- ---------- ---------
  1    17      679163     08-SEP-09 679230     08-SEP-09

BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ---------------
1025    489.00K    DISK        00:00:01     08-SEP-09
        BP Key: 1028 Status: EXPIRED Compressed: NO Tag: TAG20090908T203418
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T203418_5bg51c2c_.bkp
```

```
List of Archived Logs in backup set 1025
Thrd Seq     Low SCN    Low Time  Next SCN    Next Time
---- ------- ---------- --------- ---------- ---------
1    18      679230     08-SEP-09 679716     08-SEP-09
```

这个报告中的最后一个归档日志备份集的状态为 EXPIRED，其他的归档日志备份集的状态为 AVAILABLE。因此，所有可用的归档的重做日志备份集能够用于 RMAN 的恢复操作，而最后一个归档的重做日志备份集不能用于 RMAN 的恢复操作。如果只想查看到期的备份集，则可以添加 expired 关键字，如可以执行 list expired backup of archivelog all 命令。

8. 列出控制文件和服务器参数文件备份

还可以列出控制文件和服务器参数文件备份。list backup of controlfile 命令可以向用户提供控制文件备份的列表，而 list backup of spfile 命令可以提供服务器参数文件备份的列表。下面是这两个命令的输出示例：

```
RMAN> list backup of controlfile;

List of Backup Sets
===================
BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
774     Full     9.67M      DISK        00:00:01     08-SEP-09
        BP Key: 776 Status: AVAILABLE Compressed: NO Tag: TAG20090908T202724
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062444_5bg4ndym_.bkp
  Control File Included: Ckp SCN: 679035  Ckp time: 08-SEP-09

BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
928     Full     9.67M      DISK        00:00:01     08-SEP-09
        BP Key: 930 Status: AVAILABLE Compressed: NO Tag: TAG20090908T202923
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062563_5bg4r47m_.bkp
  Control File Included: Ckp SCN: 679329      Ckp time: 08-SEP-09

BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
1062    Full     9.67M      DISK        00:00:01     08-SEP-09
        BP Key: 1064 Status: AVAILABLE Compressed: NO Tag: TAG20090908T203421
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062861_5bg51g1m_.bkp
  Control File Included: Ckp SCN: 679822 Ckp time: 08-SEP-09
```

```
RMAN> list backup of spfile;

List of Backup Sets
===================
BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
774     Full    9.67M      DISK        00:00:01     08-SEP-09
        BP Key: 776 Status: AVAILABLE Compressed: NO Tag: TAG20090908T202724
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062444_5bg4ndym_.bkp
  SPFILE Included: Modification time: 08-SEP-09
  SPFILE db_unique_name: ROB1

BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
928     Full    9.67M      DISK        00:00:01     08-SEP-09
        BP Key: 930 Status: AVAILABLE Compressed: NO Tag: TAG20090908T202923
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062563_5bg4r47m_.bkp
  SPFILE Included: Modification time: 08-SEP-09
  SPFILE db_unique_name: ROB1

BS Key  Type LV  Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ ---------------
1062    Full    9.67M DISK              00:00:01     08-SEP-09
        BP Key: 1064 Status: AVAILABLE Compressed: NO Tag: TAG20090908T203421
        Piece Name:
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062861_5bg51glm_.bkp
  SPFILE Included: Modification time: 08-SEP-09
  SPFILE db_unique_name: ROB1
```

在这里也可以使用 list expired backup of archivelog all 命令。同样，我们可以设定时间或日志序号来限制报告中显示的内容。例如，可以使用 list expired backup of archivelog until sequence 命令列出到给定序列号为止的到期归档日志备份。

17.1.3　列出映像副本

正如使用 list 命令可以判断备份集的状态那样，也可以使用 list 命令来判断数据库映像副本的状态。如下所示，执行 list copy 命令可以生成所有副本的列表：

```
RMAN> list copy;
starting full resync of recovery catalog
full resync complete
specification does not match any control file copy in the repository
List of Datafile Copies
```

```
======================
Key     File S Completion Time Ckp SCN    Ckp Time
------- ---- - -------------- ---------- --------------
1215    3    A  08-SEP-09       681024     08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/datafile
/o1_mf_undotbs1_5bg6sw1s_.dbf
         Tag: TAG20090908T210427

List of Archived Log Copies for database with db_unique_name ROB1
=====================================================================
Key     Thrd Seq     S Low Time
------- ---- ------- - ---------
54      1    9       X 08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_9_5bd8qv45_.arc
1170    1    19       A 08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_19_5bg61141_.arc
```

除了概述列表外，还可以创建单个数据文件副本、归档的重做日志副本和控制文件副本的列表。下面将详细介绍。

1. 列出数据文件副本

Oracle 允许用户执行 list copy of database 命令生成所有数据文件副本的概述列表，如下所示：

```
RMAN> list copy of database;

List of Datafile Copies
=======================
Key     File S Completion Time Ckp SCN Ckp Time
------- ---- - -------------- ---------- --------------
1215    3    A  08-SEP-09       681024     08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/datafile
/o1_mf_undotbs1_5bg6sw1s_.dbf
         Tag: TAG20090908T210427
```

这个输出中存在属于数据库的两个数据文件副本：datafile2.copy 和 datafile3.dbf。报告中没有列出实际的数据文件名和指定的表空间名，而是在第二列列出了文件号。我们可以通过运行 report schema 命令将这个文件号和关联的表空间联系起来，本章后面部分将讨论这个问题。

如果要知道是否存在表空间或数据文件的数据文件副本，则可以使用 list copy of tablespace 或 list copy of datafile 命令，如下所示。

```
RMAN> list copy of tablespace undotbs1;

List of Datafile Copies
=======================
Key     File S Completion Time Ckp SCN    Ckp Time
------- ---- - --------------- ---------- ---------------
1215    3    A 08-SEP-09       681024     08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/datafile
/o1_mf_undotbs1_5bg6sw1s_.dbf
         Tag: TAG20090908T210427
```

2. 列出归档的重做日志副本

如果要列出归档的重做日志副本，可以使用 list copy of archivelog all 命令，如下所示：

```
RMAN> list copy of archivelog all;
List of Archived Log Copies for database with db_unique_name ROB1
=====================================================================
Key     Thrd Seq     S Low Time
------- ---- ------- - ---------
54      1    9       X 08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/
2009_09_08/o1_mf_1_9_5bd8qv45_.arc

1170    1    19      A 08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_19_5bg6114l_.arc
```

我们也可以按照时间、序列或数据库 SCN 列出特定的归档的重做日志副本。下面是一些基于不同标准列出归档的重做日志副本的示例：

```
RMAN> list copy of archivelog from sequence 9;

List of Archived Log Copies for database with db_unique_name ROB1
=====================================================================
Key     Thrd Seq     S Low Time
------- ---- ------- - ---------
54      1    9       X 08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_9_5bd8qv45_.arc

1170    1     19     A 08-SEP-09
        Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_19_5bg6114l_.arc
```

```
RMAN> list copy of archivelog from sequence 9 until sequence 19;
List of Archived Log Copies for database with db_unique_name ROB1
=====================================================================
Key     Thrd Seq     S Low Time
------- ---- ------- - ---------
54      1    9       X 08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_9_5bd8qv45_.arc

1170    1    19        A 08-SEP-09
        Name:
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_19_5bg6114l_.arc
```

3. 列出控制文件副本

RMAN 还可以通过执行 list copy of controlfile 命令来报告控制文件副本，如下所示：

```
RMAN> list copy of controlfile;
List of Control File Copies
===========================
Key     S Completion Time Ckp SCN    Ckp Time
------- - --------------- ---------- ---------------
1795    A 08-SEP-09       682922     08-SEP-09
         Name:
/oracle/app/oracle/flash_recovery_area/ROB1/controlfile
/o1_mf_TAG20090908T213230_5bg8gh41_.ctl
         Tag: TAG20090908T213230
```

17.2 RMAN 的 report 命令

RMAN 的 report 命令用于判断数据库的当前可恢复状态和提供数据库备份的特定信息。本节将介绍显示在指定时间周期内哪些数据文件还是未备份的报告，还将介绍显示何时因数据文件上的 UNRECOVERABLE 操作而需要备份特定表空间的报告，最后再介绍如何使用 report 命令来报告数据库模式和丢弃的数据库备份。

17.2.1 报告最近没有被备份的数据文件

DBA 经常要弄清楚这样一个问题："最后一次备份这个表空间是在什么时间"？通过 report need backup 命令可以得到一些这样的信息。例如，如果想知道过去 3 天内有没有备份过的表空间，可以执行 report need backup day=3 命令来查找这些表空间，下面是执行该命令的输出示例：

```
RMAN> report need backup days=3;
```

```
Report of files whose recovery needs more than 3 days of archived logs
File  Days Name
---- ----- -----------------------------------------------------
4     2     D:\ORACLE\ORADATA\RECOVER\TOOLS01.DBF
5     2     D:\ORACLE\ORADATA\RECOVER\USERS01.DBF
```

这个报告中的两个数据文件要求应用 3 天以上的归档的重做日志才能被恢复(也就是说最近 3 天没有备份这两个文件)。此时，最好备份这些数据文件及其关联的表空间。

还可以根据需要应用的增量备份数生成报告，如下所示：

```
RMAN> report need backup incremental = 3;

Report of files that need more than 3 incrementals during recovery
File Incrementals Name
---- ------------ -----------------------------------------------
1    4            D:\ORACLE\ORADATA\RECOVER\SYSTEM01.DBF
2    4            D:\ORACLE\ORADATA\RECOVER\RECOVER_UNDOTBS_01.DBF
3    4            D:\ORACLE\ORADATA\RECOVER\INDX01.DBF
4    4            D:\ORACLE\ORADATA\RECOVER\TOOLS01.DBF
5    4            D:\ORACLE\ORADATA\RECOVER\USERS01.DBF
```

在这个示例中，一些数据库数据文件要求应用 4 个增量备份，这说明需要对这些数据文件执行更高级别的增量备份操作，或者需要执行新的增量基本备份操作。

17.2.2　报告备份冗余或恢复窗口

我们可以执行 report need backup redundancy 命令来确定为满足备份冗余策略而需要备份的数据文件。下面将介绍一个这样的报告示例。在这个示例中，我们要列出所有不包含至少两个不相同备份的数据文件，这些备份可能是备份集备份或数据文件副本。

```
RMAN> report need backup redundancy = 2;
Report of files with less than 2 redundant backups
File #bkps Name
---- ----- -----------------------------------------------
1    1     D:\ORACLE\ORADATA\RECOVER\SYSTEM01.DBF
4    1     D:\ORACLE\ORADATA\RECOVER\TOOLS01.DBF
5    1     D:\ORACLE\ORADATA\RECOVER\USERS01.DBF
```

同样地，我们可以执行 report need backup recovery window days 命令为备份建立一个时间值最小的恢复窗口，并且报告备份时间长于这个恢复窗口的数据文件，如下所示：

```
RMAN> report need backup recovery window of 2 days;

Report of files whose recovery needs more than 2 days of archived logs
File Days  Name
---- ----- -----------------------------------------------------
1    4     D:\ORACLE\ORADATA\RECOVER\SYSTEM01.DBF
2    4     D:\ORACLE\ORADATA\RECOVER\RECOVER_UNDOTBS_01.DBF
3    4     D:\ORACLE\ORADATA\RECOVER\INDX01.DBF
```

```
4     4      D:\ORACLE\ORADATA\RECOVER\TOOLS01.DBF
5     4      D:\ORACLE\ORADATA\RECOVER\USERS01.DBF
```

在这个示例中，一些数据文件要求应用两天以上的归档的重做日志。因此，如果恢复策略要求备份只需应用一天的重做日志，就必须备份这些数据文件。

17.2.3 报告数据文件的不可恢复操作

对表空间中的对象与组成这些表空间的数据文件执行的不可恢复操作会导致某些可恢复性问题。例如，如果使用 Unrecoverable 选项创建了一个表，随后使用直接加载路径加载这个表，就需要备份这个表空间，否则加载的数据不可恢复。此时，可以执行 report unrecoverable 命令，如下所示：

```
RMAN> report unrecoverable;
Report of files that need backup due to unrecoverable operations

File  Type of Backup Required Name
---- ----------------------- -----------------------------------
5     full or incremental     D:\ORACLE\ORADATA\RECOVER\USERS01.DBF
```

17.2.4 报告数据库模式

这里所使用的术语“模式(schema)”指的是数据库的物理结构。模式包括数据文件名、数据文件号、为这些数据文件指派的表空间、数据文件的大小，以及数据文件是否含有回滚段。我们可以报告当前的模式，也可以报告过去时间点的数据库模式。下面是一个执行 report schema 命令的示例：

```
RMAN> report schema;

    Report of database schema for database with db_unique_name ROB1
    List of Permanent Datafiles
    ===========================
    File Size(MB) Tablespace           RB segs Datafile Name
    ---- -------- -------------------- ------- ------------------------
    1    700      SYSTEM               YES
/ora01/oracle/rob1/rob1/system01.dbf
    2    600      SYSAUX               NO
/ora01/oracle/rob1/rob1/sysaux01.dbf
    3    280      UNDOTBS1             YES
/ora01/oracle/rob1/rob1/undotbs01.dbf
    4    8        USERS                NO
/ora01/oracle/rob1/rob1/users01.dbf
```

17.2.5 报告过时的备份

如果使用了保留策略(第 14 章介绍过)，备份有可能会被标记为过时状态(OBSOLETE)。下面是一个执行 report obsolete 命令的示例，此时备份策略被设置为 redundancy 1。

```
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          54     08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/archivelog/2009_09_08
/o1_mf_1_9_5bd8qv45_.arc
Backup Set           712    08-SEP-09
  Backup Piece       714    08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_annnn_TAG20090908T202600_5bg4kr90_.bkp
Backup Set           774    08-SEP-09
  Backup Piece       776    08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062444_5bg4ndym_.bkp
Backup Set           928    08-SEP-09
  Backup Piece       930    08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062563_5bg4r47m_.bkp
Backup Set           1062   08-SEP-09
  Backup Piece       1064   08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697062861_5bg51g1m_.bkp
Backup Set           1259   08-SEP-09
  Backup Piece       1262   08-SEP-09  /oracle/app/oracle/flash_
recovery area/ROB1/ autobackup/2009_09_08/o1_mf_s_697064685_5bg6tfxj_.bkp
Backup Set           1413   08-SEP-09
  Backup Piece       1416   08-SEP-09    /oracle/app/oracle/flash
recovery area/ROB1/backupset/2009_09_08
/o1_mf_ncnnf_TAG20090908T212707_5bg84dxo_.bkp
Backup Set           1475   08-SEP-09
  Backup Piece       1477   08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697066032_5bg84jf9_.bkp
Backup Set           1598   08-SEP-09
  Backup Piece       1601   08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_08
/o1_mf_ncnnf_TAG20090908T213150_5bg8f7q1_.bkp
Control File Copy    1795   08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/controlfile
/o1_mf_TAG20090908T213230_5bg8gh41_.ctl
Backup Set           1664   08-SEP-09
  Backup Piece       1666   08-SEP-09
/oracle/app/oracle/flash_recovery_area/ROB1/autobackup/2009_09_08
/o1_mf_s_697066314_5bg8fbx9_.bkp
```

这个报告中含有被 Oracle 标记为 OBSOLETE 的一些不同的备份集、数据文件副本、

控制文件副本和归档日志副本。如果要标记这些备份的状态为 DELETED，则可以使用第 14 章介绍过的 delete obsolete 命令。

17.3　用于报告的数据字典视图

Oracle 提供了一些与 RMAN 有关的数据字典视图(v$views)，可通过 SQL 提示符利用该类视图执行报告。可以使用这些视图创建自定义报告。按照获得的报告完成相应的工作，如在数据库未备份或未在恢复目录中注册数据库时发出通知(可以使用一些易于做比较的配置控制)。所有与 RMAN 有关的 Oracle 视图可从 *Oracle Reference Guide* 中获取(包括视图的目的以及各列的含义等)。多数 RMAN 视图以 V$BACKUP*, V$RECOVERY* 及 V$RMAN 为前缀。表 17-1 列出了常用的视图。

表 17-1　用于报告的数据字典视图

视 图 名	描　述
V$BACKUP_ARCHIVELOG_DETAILS	包含所有可还原归档日志的信息
V$BACKUP_ASYNC_IO	提供正在进行的和最近完成的RMAN备份和还原的性能信息
V$BACKUP_CONTROLFILE_DETAILS	提供可还原控制文件的信息
V$BACKUP_COPY_DETAILS	提供可用的控制文件和数据文件副本的相关信息
V$BACKUP_CORRUPTION	提供从控制文件获得的关于数据文件备份损坏块的相关信息
V$BACKUP_DATAFILE	提供从控制文件获得的关于备份集中的数据文件和控制文件的信息
V$BACKUP_DATAFILE_DETAILS	提供可还原的数据文件的相关信息
V$BACKUP_FILES	提供所有的 RMAN 备份(包括映像副本和备份集)的信息
V$BACKU_PIECE	提供从控制文件获得的备份片的信息
V$BACKU_PIECE_DETAILS	提供所有可用的备份片的信息
V$BACKU_REDOLOG	提供从控制文件获得的在备份集中的归档日志的信息
V$BACKU_SET	提供从控制文件获得的备份集的信息
V$BACKU_SET_DETAILS	提供从控制文件获得的备份集的详细信息
V$BACKU_SPFILE	提供从控制文件获得的在备份集中的服务器参数文件的信息
V$BACKU_SPFILE_DETAILS	提供从控制文件获得的在备份集中的服务器参数文件的详细信息
V$BACKU_SYNC_IO	提供正在进行的和最近完成的RMAN备份和还原的性能信息

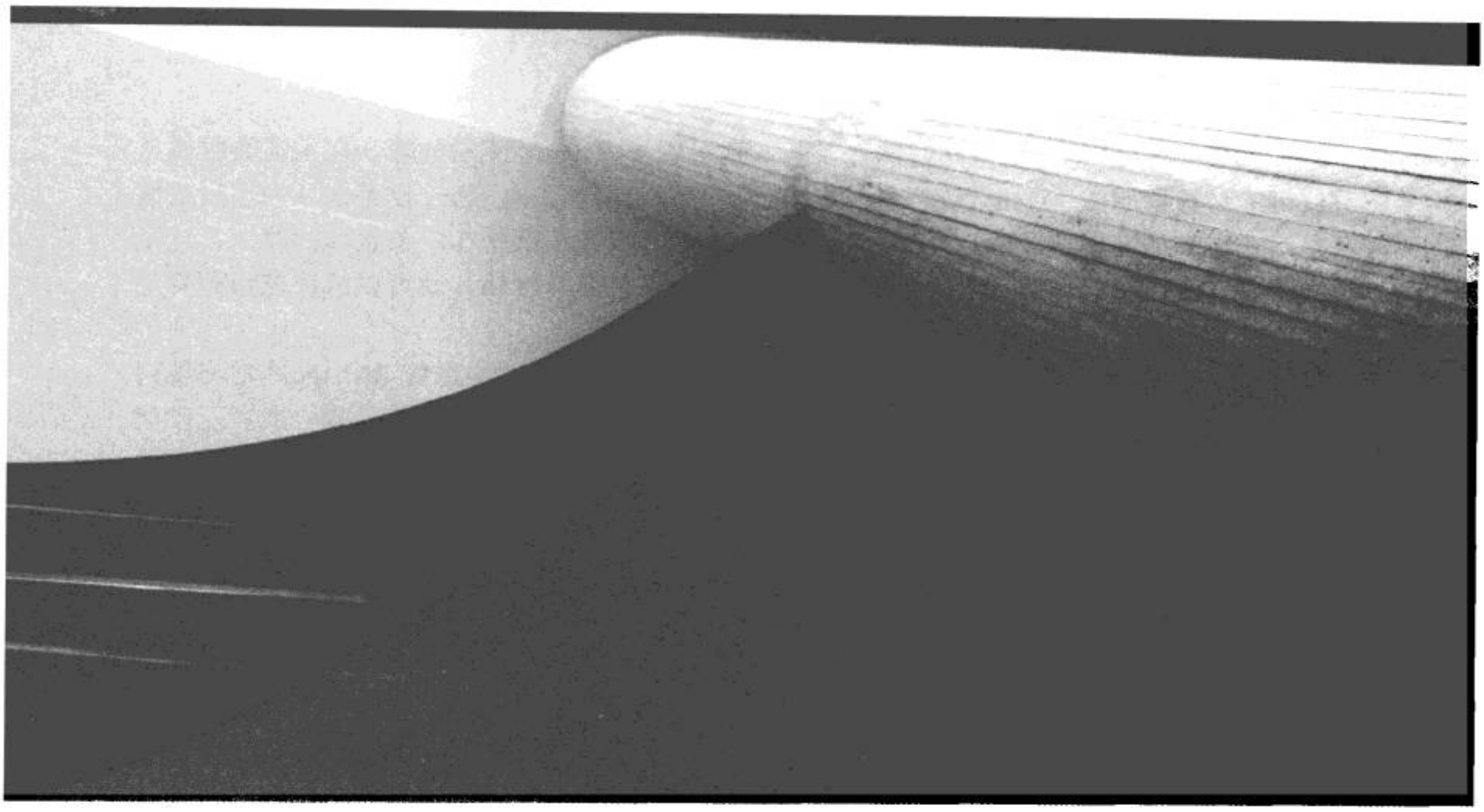
(续表)

视　图　名	描　　述
V$DATABASE_BLOCK_CORRUPTION	提供上一次备份后损坏的数据块的信息
V$RECOVER_FILE	包含需要进行介质恢复的文件的状态信息
V$RECOVER	提供与恢复区有关的信息
V$RMAN_BACKUP_JOB_DETAILS	提供备份工作的详细信息
V$RMAN_BACKUP_TYPE	提供 RMAN 备份类型的信息
V$RMAN_CONFIGURATION	提供 RMAN 的持久性配置设置有关的信息
V$RMAN_OUTPUT	提供由 RMAN 报告的消息
V$RMAN_STATUS	提供 RMAN 完成的和正在进行的工作的信息

多数RMAN的控制文件的视图与恢复目录的视图等价，恢复目录中的相关视图以RC_为前缀，而不是以 V$为前缀。其视图的名称相似。例如，V$ARCHIVED_LOG 视图提供控制文件中的归档重做日志，而 RC_ARCHIVED_LOG 视图提供了源自恢复目录的同样的视图。最后，可以发现在 Oracle 中多数与性能有关的视图可用于帮助 RMAN 操作的性能调整工作。在第 18 章中将更加详细地讨论如何使用那些视图。

17.4　小结

本章提供了用于从 RMAN 中析取信息的命令。list 和 report 命令提供了用于管理 RMAN 和确保获得数据库的良好备份的丰富信息。我们应该确定对数据库生成哪些类型的报告，并且自动化这些生成报告的过程，从而随时了解数据库的备份和恢复状态。



第 18 章

调整 RMAN 备份与恢复操作的性能

RMAN 实际上是即装即用的，我们通常不需要对其做什么调整。不过，RMAN 体系结构中还包含许多组件，当这些组件构成一个整体时，就必须调整 RMAN 的设置以从备份进程中得到最佳的性能。通常，RMAN 调整涉及处理逻辑或物理数据库设计中的低效率、调整介质管理库(Media Management Library，MML)、调整 RMAN 和 MML 层以与备份数据的物理设备更好地共存。

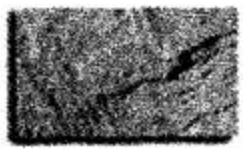
如果使用过 Oracle Database 10g 之前的 RMAN，您就会发现与性能相关的命令已有了一些变化。根据我们的估计，这些变化使得调整过程更容易理解。本章将介绍在开始调整 RMAN 前需要完成的调整工作，随后还要介绍为 RMAN 提供的一些调整选项。

18.1 调整 RMAN 前的工作

如果 RMAN 的备份操作时间非常长，这可能不是 RMAN 的故障。在大多数时候，这可能是数据库或 MML 存在问题。就好比在高峰期间行车，您一定不会认为速度过慢是由于汽车的问题造成的。实际上，问题可能是有很多车都挤在一条狭窄的路上，这是一个带宽或瓶颈问题。通过拓宽公路或建设地铁、公交、轻轨，就可以解决城市高峰期间的交通问题。

调整 RMAN 和备份与恢复进程时，也会出现同样的问题。尽管 RMAN 经常出现问题，但是这并不一定是 RMAN 的故障。问题很可能出在系统总的带宽不够，或者基础架构中的某些组件配置不正确，RMAN 总是第一个被指责，而 RMAN 只是造成这种局面的一个因素而已。

一旦体系结构配置且操作正确，大多数 RMAN 调整实际上就是 Oracle 数据库的调整。数据库的性能越好，RMAN 备份操作的性能就越好。鉴于有很多书籍都讨论过这个主题，因此我们只作一些简要介绍。如果要得到有关调整 Oracle 数据库性能的详细信息，我们建议您阅读清华大学出版社引进并出版的《Oracle Wait Interface 性能诊断与调整实践指南》。



注意：
本章以及本书的许多地方都提供了一些调整建议。确保在决定“激活并忘记”(意味着进行更改，而不检查这种更改是否是正确的)之前在系统上测试这些建议。由于所指定的配置可能不适用于您所处的环境，因此在按照建议执行各种操作前，您必须测试这些建议。

18.1.1 可以达到的 RMAN 性能

在当前有效的技术条件下，RMAN 性能可以提高到什么水平？Oracle 公司在它的白皮书 *Oracle Recovery Manager: Performance Testing at Sun Customer Benchmark Center* (www. oracle.com/technology/index.html 网站上有提供)中认为：在磁带上备份和恢复数据的速率能够达到每小时 1TB。随着磁带备份技术的不断发展，这个速率还可能变得更大。

18.1.2 使用合适的硬件

如果想得到更高的备份性能，首先要考虑的是所配置的备份硬件。备份硬件可能包括磁带设备及其关联的基础架构(如电缆、自动磁带接口和可能选用的 MML 层软件)。

备份介质硬件决定了读写设备的速度。显然，写设备的速度越快，备份的速度就会越快。此外，执行备份操作的设备越多，备份速度也会越快。在前面提到的 Oracle 的 RMAN 性能白皮书中已经明确说明了这个问题。增加 RMAN 写入的驱动器几乎能够线性地提高备份和还原操作的效率。在多个信道或备份设备中并行执行备份操作的能力对

于快速备份大型 Oracle 数据库是至关重要的。

RMAN 能够受益于并行的 CPU 资源，但是仅限于此而已，再增加更多的 CPU 并不会继续使性能显著提高，这与使用更多的物理备份设备是完全不同的。一般说来，在大多数情况下，使用多个备份设备比添加 CPU 更能够对备份和还原窗口产生积极的影响。

您会发现：大多数备份设备是异步的而不是同步的。异步设备使得备份服务器进程无须等待 I/O 的完成就可以发出 I/O 指令。例如，一个异步操作允许服务器进程发出磁带写指令，在执行这个指令的同时，该进程可以继续填充内存缓冲区以准备下一个写操作。另一方面，同步设备在执行其他任何工作前都必须等待备份操作的完成。在我们要介绍的示例中，同步进程在为下一个操作启动内存缓冲区的填充前必须等待磁带 I/O 的完成。因此，异步设备比同步设备更有效率。

由于异步操作是首选的，所以您需要了解与异步操作有关的一些参数。首先，BACKUP_TAPE_IO_SLAVES 参数(默认值为 FALSE)会使所有磁带 I/O 实际上是异步的。如果支持磁带设备的异步 I/O，我们建议您将这个参数值设置为 TRUE(如果支持该设置)，以启动该设置。一旦建立 BACKUP_TAPE_IO_SLAVES 参数后，就可以使用 allocate channel 命令或 configure channel 命令的 parms 参数来定义内存缓冲区的大小。

磁带缓冲区的大小是在配置信道时确定的，它的默认值是由操作系统决定的，不过通常为 64KB。使用 allocate channel 命令可以将磁带缓冲区的大小设置为不同的值。为了达到最佳性能，我们建议您将磁带缓冲区的大小设置为 256KB 或更大，如下所示：

```
allocate channel c1 device type
   sbt parms="blksize=262144, ENV=(NB_ORA_CLASS=RMAN_db01)"
```

如果要在磁盘上备份数据，我们必须判断操作系统是否支持异步 I/O。如果支持，Oracle 会自动使用异步 I/O 的功能；如果不支持，此时将 Oracle 提供的 DBWR_IO_SLAVES 参数设置为非零值，通过启动多个 DBWR 进程，Oracle 会模拟到磁盘的异步 I/O。

当配置 DBWR_IO_SLAVES 或 BACKUP_TAPE_IO_SLAVES 时，可能也需要创建一个大池。这将帮助消除共享池争用和内存分配错误的问题，这些是在启用 BACKUP_TAPE_IO_SLAVES 时伴随共享池使用一起发生的问题。如果正在使用 Automatic Shared Memory Management(ASMM)，Oracle 就将管理共享池的内存分配。如果需要手动设置大池，则磁盘缓冲区的总大小限制为每个信道 16MB。为备份设置 LARGE_POOL_SIZE 参数的公式如下：

```
LARGE_POOL_SIZE = (number of allocated channels) * (16MB + (4*size of tape
   buffer))
```

注意：

如果没有配置 DBWR_IO_SLAVES 或 BACKUP_TAPE_IO_SLAVES，RMAN 就不会使用大池。一般来说，除非操作系统不支持异步 I/O，这时才需要配置这些参数设置以从 RMAN 中获得良好的性能。

18.1.3 调整数据库

欠缺良好调整的数据库对备份时间将会产生非常消极的影响。某些数据库的调整问题也会显著地影响还原时间。本节简要介绍这些调整问题，包括 I/O 调整、内存调整以及 SQL 调整。

1. 调整 I/O

大多数 DBA 都会理解 I/O 对基本的数据库操作的影响。争用给定磁盘驱动器上的数据库资源(如位于相同设备上的联机重做日志和一个数据库数据文件)会显著降低系统运行的速度。较差的 I/O 分发不但会影响数据库的性能，还会影响备份和还原时间，这是因为 RMAN 就是在设备上争夺 I/O 时间的另一个进程(或者是并行的许多进程)。

备份是一个读密集型(read-intensive)操作。如果 I/O 分发较差，不仅 RMAN 性能会受到影响，而且用户会在备份操作过程中受到影响。如果所有恢复都是完全数据库恢复，恢复操作可能会简单一些。但是，如果只恢复一个数据文件或一个表空间，由于数据库被打开并处于使用状态，就能够发现较差的 I/O 分发会影响恢复窗口和用户。从根本上说，差的 I/O 分发不仅会影响日常的数据库用户，而且会导致备份和恢复操作花费更长的时间。

有关 Oracle 数据库的 I/O 分发问题的资料很多，我们建议您阅读 Oracle 公司的白皮书 *Oracle Storage Configuration Made Easy*，在 www.oracle.com/technology/index.html 网站上可以查找到这份资料。在这个白皮书中，Mr.Loaiza 提出了一个称为 SAME(Stripe and Mirror Everything)的 I/O 分发观点，讨论了当前磁盘速度和流量，然后论证了 SAME 方法的逻辑性。SAME 方法建议我们在尽可能最大数目的多个磁盘上对数据分条，并认为这种方法比使用几个磁盘或磁盘分区对数据分条更好，例如 RAID-5(当然，镜像更为昂贵)。此外，这个白皮书认为分条数据的尺寸为 1MB 时通常是最佳的，并且证明 Oracle 的这种配置在磁盘中读/写数据的速度比没有对数据分条的系统的读/写速度快 13%，这样就节省了备份时间。较快的磁盘读/写将转换为较快的备份时间。

2. 调整内存的使用

与所有的 Oracle 进程相同，RMAN 要使用内存。启动一个 RMAN 操作时，就会为这个 RMAN 操作分配一个用于工作的缓冲区，我们可以根据下面这些因素确定这个缓冲区的大小：

- RMAN 备份和恢复多路复用的影响
- 所使用的设备类型
- 操作期间所分配的信道数

上面的每一种因素都会影响 RMAN 所需内存的大小。RMAN 为各种操作分配内存缓冲区，分配这些内存缓冲区的方法取决于要使用的设备类型。接下来将详细介绍不同的内存缓冲区分配方法。

为磁盘设备分配内存缓冲区

在磁盘设备上备份数据时，RMAN 最多可以分配 16MB 内存，这个内存是基于多路复用级别(多路复用级别是由 filesperset 参数设置的)分配的。如果多路复用级别为 4 或以下，RMAN 会分配每个大小为 1MB 的 16 个缓冲区。这些 1MB 缓冲区在要备份的数据文件数之间分配。因此，如果 filesperset 参数被设置为 2，每个数据文件就会被分配 8 个 1MB 的缓冲区。

如果 filesperset 参数被设置为 5 到 8 之间的数，RMAN 就会分配大小为 512KB 的缓冲区，并且均匀地分发给不同的数据文件，此时，所分配的内存缓冲区不超过 16MB。如果多路复用级别大于 8，每个数据文件会被分配 4 个 128KB 的缓冲区，为每个数据文件分配的缓冲区总计为 512KB。

为 SBT 设备分配内存缓冲区

在 SBT 设备上备份数据时，RMAN 会为所分配的每个信道分配 4 个缓冲区。这些缓冲区的大小通常为 256KB，因此为每个信道分配的总内存为 1MB。可以使用 allocate 或 send 命令的 PARMS 和 BLKSIZE 参数管理缓冲区大小。

这个内存通常是从 PGA 中分配的，不过如果 BACKUP_TAPE_IO_SLAVES 参数被设置为 TRUE，除非分配了大池，否则就需要使用 SGA，在这种情况下要使用大池。因此，如果要配置 I/O 从属(在 SBT 设备上备份数据时通常应当配置 I/O 从属)，就应当配置一个大池，以减少大池对整个内存的需求。

3. 调整 SQL

您一定会问：为什么在数据库中运行 SQL 会影响对备份和恢复时间的调整？很明显，较差的 SQL 语句操作会对整个数据库和数据库所在的系统产生消极影响。任何对数据库的消极影响同样会对备份操作产生消极影响。我们应当调整 SQL 操作，以减少所需的 I/O 数(逻辑上的和物理上的)，并安排在系统空闲期间执行备份操作(如果可能)。

4. 调整环境

仔细查看备份调度，确认这些调度不与 I/O 密集型数据库操作(如数据加载或报告)冲突。另外，如果您发现备份操作的时间过长，就需要考虑使用增量备份策略，同时分析数据库并判断是否存在只读的表空间，这样就不需要经常备份只读的表空间。此外，如果在 ARCHIVELOG 模式中运行数据库，我们也可以考虑在不同日期交错备份表空间，以减少整个备份操作的时间(当然这是以更长的恢复时间为代价)。

如果使用 Oracle Real Application Clusters 运行数据库，RMAN 会充分利用集群的环境来并行化 RMAN 操作(第 19 章将深入讨论这个问题)。

5. 调整备份和恢复策略

前面介绍了执行备份操作时间的调整，这部分将要讨论调整执行备份操作的方法。如果数据库相当大，我们就不能只执行一条 backup database 命令来备份全部文件集，而

应当使用分块备份策略。可以考虑使用单独的 backup tablespace 或 backup datafile 命令对可能需要一起还原的相关数据文件进行备份，并且为这些备份分配一个特定的信道，这样可以减少在恢复操作期间交换磁带的次数，还可以最快地恢复相关的数据文件。下面是这个操作的示例：

```
RMAN> backup (tablespace tools channel 'ORA_DISK_1')
             (tablespace system, undotbs, users, testrbs, indx);
```

如果要备份一个稍后可能需要恢复的只读表空间，或者要经常对一个表空间执行时间点恢复操作(TSPITR)，就可以执行这样的操作。上述情况都可以被选作备份策略。

Oracle Database 11g提供了一种新功能用于调整备份与恢复策略。以前版本的RMAN在备份时，给定的数据文件只能通过一个信道备份。这也就意味着，如果在备份环境中，存在多个较小的数据文件和一两个较大的数据文件，每个备份集规则采用单一的备份信道时，备份或还原需要花费大量的时间。而采用多段备份功能时，可以拓展单一的数据文件 I/O 信道，使其可利用多信道 I/O，从而大大提高备份进程的速度。

另一个需要考虑的方面是备份策略对恢复操作的影响。其中，RMAN 难以解决的一个问题是：如果要使用 restore database 命令还原一个完整的数据库，根据所使用的平台(以 UNIX 平台为例，接下来的三段文字对在 Windows NT 上运行的 Oracle 不适用)，必须保证有足够的空间能够用于这个还原操作。只要有足够的磁盘空间，恢复操作就不会有问题。但想一想磁盘空间不足的灾难恢复情况，在这种情况下，RMAN 会花费数小时来恢复数据库，一旦用完了磁盘空间，您可能会认为 RMAN 只是停下来，发出缺少空间的警告，然后停止还原操作。但实际情况会更糟糕。

还原进程失败时，RMAN 会删除在这个还原会话中还原的每个不完整的文件。因此，如果用户花费两个小时还原了除一个数据库数据文件之外的所有文件，而此时磁盘空间耗尽，由于 RMAN 要删除所有已还原的文件，所以就会前功尽弃，这对于 DBA 来说是不能接受的。

解决这个问题的一个方案是对每个表空间(或几个表空间)都使用不同的 restore 命令，这样就可以以更小的粒度还原数据库数据文件，即使其中一个还原操作失败也不会引起更大的损失。当然，如果要使用这种操作类型，就应当并行化还原操作，并且利用多个磁盘或磁带，以减少争用和磁带流量问题。此时，需要运行并行的 RMAN 会话，每个 RMAN 会话还原各自的一组表空间数据。需要注意的是，这种还原操作类型也是为什么要在同一个信道上备份还原所需的文件集群的另一个重要原因，这样我们就可以在磁带上快速地读取数据。

需要注意的是，某些平台(如 Windows NT)会在启动 RMAN 的数据库、数据文件和表空间还原操作前检查可用的空间，这样在出现空间不够的情况时就不会浪费还原的时间。

18.2 调整 RMAN

本章开始部分介绍过，即装即用的 RMAN 的运行相当稳定，不过仍然可以调整

RMAN 以达到更佳的性能，这是本节要介绍的内容。我们首先要讨论 RMAN 中的调整选项，然后还要讨论一些 MML 调整问题。

18.2.1　调整 RMAN 设置

本小节将介绍一些调整 RMAN 的方法。调整 RMAN 涉及调整并行操作和配置 RMAN 使用多路复用(或不使用多路复用)。此外我们还将介绍其他一些降低 RMAN 读写数据速率的操作。

1. 并行信道操作

我们在调整数据库备份时可能产生的最大影响是使用多个 RMAN 信道并行化这些数据库备份。通常，我们为要备份数据的每个设备都配置一个信道。因此，如果要在 3 个不同的磁盘上备份数据，就需要配置 3 个不同的信道。在同一个设备上很少存在并行化的备份操作，如果在 Windows NT 系统上存在 D 驱动器和 E 驱动器，并且这两个驱动器是同一个磁盘的不同分区，那么在这两个驱动器上执行并行化的备份操作并不能提高速度。

注意：
磁带系统上的内存缓冲可能会使得额外分配的信道更有价值，因此我们应当经常执行一些时间测试，以确定在系统中需要使用的信道数。

为备份分配多个信道可以实现备份操作的并行化。第 3 章曾经讨论过，使用 configure 命令可以为信道配置默认值(以及决定使用多少个信道的默认并行级别)。例如，如果要在两个磁带设备上备份数据，就可以配置两个默认的信道和一个默认的并行级别，如下所示：

```
CONFIGURE DEVICE TYPE DISK PARALLELISM 2;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt FORMAT PARMS
'ENV=(NB_ORA_CLASS=RMAN_rs100_tape)';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt FORMAT PARMS
'ENV=(NB_ORA_CLASS=RMAN_rs100_tape)';
```

上面的命令将确保每条 backup 或 recover 命令都自动分配两个信道来并行化备份或恢复进程。

此外，根据所使用的设备，我们还能够在磁盘上运行并行的数据流。最好能运行一些测试，并根据测试结果从备份和恢复的角度观察不同数目的备份数据流对性能的影响。在备份操作期间和在恢复操作期间使用不同数目的数据流也许会更加有效率，必须经过测试才能确定数据流的数目。

前面已经讨论过 RMAN 中的备份多路复用以及调整这个功能的方法。调整备份多路复用会显著地影响备份和恢复性能。我们在前面还介绍过多路复用的配置如何对为 RMAN 备份进程分配的总内存产生影响。通常，只要不会导致来回的磁盘交换，内存越大，性能就会越佳。

此外，适当地配置多路复用会更有效地在磁带上传递数据流。为 RMAN 分配的内存

越大，I/O 设备上传输的数据就越多。对于新型的磁带驱动器来说，磁带数据流不再是一个问题。这些新型的磁带驱动器通常使用大量的缓冲区内存以确保以不变的速率在磁带上写入数据。

2. RMAN 多路复用

在考虑 RMAN 的性能之前，我们先讨论一下 RMAN 中的多路复用。多路复用(multiplexing)允许使用单个 RMAN 信道在备份进程期间并行读取数据库数据文件，并且将这些数据文件的内容写入同一个备份集片。因此，一个备份集片可能包含了许多不同数据文件的内容。

需要注意的是，某个特定数据文件的内容可以驻留在多个备份集片中(您可以通过将 maxpiecesize 参数设置为比任何数据文件大小还要小的一个值来证明这个现象，但是同时必须将 maxsetsize 参数设置为不小于要备份的最大表空间大小的一个值)。不过在某个特定的备份中，只能通过一个信道或备份集备份一个数据文件。因此，如果分配了两个信道(这样就具有两个备份集)，并且数据库由一个大型数据文件组成，RMAN 并行备份这个数据文件的能力会受到很大的限制。

RMAN 的多路复用级别由下列两个参数中值较小的参数决定。第一个参数是 filesperset，该参数在执行 backup 命令时确定；第二个参数是 maxopenfiles，该参数在分配信道时确定。

filesperset 参数确定每个备份集所包含的数据文件的个数。一个备份集中的数据文件个数应当小于等于 filesperset 参数值。执行备份操作时，RMAN 会为 filesperset 参数指派一个默认值，这个默认值可以为 64，也可以是输入文件数除以分配信道数的结果(取较小的值)。可以通过执行带有 filesperset 参数的 backup 命令来使用非默认的 filesperset 参数值，如下所示：

```
backup database filesperset 4;
```

maxopenfiles 参数确定 RMAN 能够并行读取的数据文件个数(默认值为 8)，该参数是基于每个信道来确定的。下面是一个使用 maxopenfiles 参数的示例：

```
CONFIGURE CHANNEL 1 DEVICE TYPE DISK MAXOPENFILES 3 FORMAT
    "d:\backup\recover\%U";
```

例如，如果将 filesperset 参数设置为 6，maxopenfiles 参数设置为 3 创建一个备份集，RMAN 就只能同时并行备份 3 个数据文件。假设分配了一个信道，并且并行化级别为 1，此时每个备份集仍然可以最多包含 6 个数据文件，但是在任何时候都只能同时写入 3 个数据文件。

注意：
如果在多个磁盘上分块备份数据，就不需要多路复用备份操作，并且可以将 maxopenfiles 参数设置为 1。如果在很少的几个磁盘上分块备份数据，可以考虑将 maxopenfiles 参数设置为 4 到 8 之间的一个值。如果不需要分块备份数据，通常应当将 maxopenfiles 参数设置为 8 以上的值。

多路复用与 filesperset 参数和 maxopenfiles 参数的使用能够显著地影响(改善或恶化)备份操作的性能。只要系统在多路复用期间具备并行操作的能力，调整 RMAN 的多路复用就可以节省备份操作所花费的时间。但是物极必反，不是说并行度越高，备份操作的性能就越佳。实际上，过分追求并行化会导致系统的工作负荷加重，在这种情况下，我们会发现系统的性能降低，而备份所花费的时间迅速增加。

多路复用也会影响磁带操作。由于磁带系统是流设备，所以为了不间断地写入数据，保持流入设备的数据流速率非常重要。通常，一旦输入磁带的数据流出现延迟，这个磁带设备会在下一个写操作前停止写操作，并且重新定位写入的头，这样就会显著地影响备份操作的总性能。通过将 filesperset 参数设置为较大值和将 maxopenfiles 参数设置为较小值，我们可以调整备份数据更有效地流入磁带设备。当然，也不能太极端，以免 I/O 信道或 CPU 不支持 RMAN 提供的数据流量。通常，应当通过调整和监控来寻求适当的平衡。

3. 控制 RMAN 操作的整体影响

在某些时候，我们需要减低 RMAN 读写数据的速率。在 Oracle Database 10g 之前，我们使用 RMAN 参数 rate 和 readrate 来抑制 RMAN 读写数据的速率、释放系统资源以用于其他操作。这些参数在为 RMAN 操作分配信道时设置。在 Oracle Database 10g 中，这些参数仍然可用，但是我们将使用其他的参数替代。

Oracle Database 10g 简化了控制 RMAN 备份的操作，现在可以在 backup 命令中使用 duration 参数来控制备份的持续时间。duration 参数有一个额外的关键字 minimize load，该关键字用于指示 RMAN 最小化在给定持续时间内备份数据库所需的 I/O 负载。例如，如果备份一般需要花费 5 个小时并占用 90%的可用 I/O，则可以指示 RMAN，该备份使用 10 小时的持续时间。当 minimize load 参数中指示这一点时，我们就可以看到只使用了 45%到 50%的可用 I/O，而不是使用了 90%的可用 I/O。当然，这种方法的负面影响是备份将花费更长的时间。下面是在开始备份时使用 duration 参数的示例。在该示例中，我们让备份运行 10 个小时：

```
Backup as copy database duration 10:00 minimize load database;
```

当然，使用 duration 参数的一个问题是备份的实际时间可能超过 10 个小时。任何完整的备份集都可用于恢复，即使备份进程由于持续时间导致的问题而失败。在超出持续时间并且备份失败时，我们可以使用 partial 关键字来取消 RMAN 错误。

使用 duration 参数时还要注意，包含最老备份的数据文件将比包含日期最新 RMAN 备份的数据文件具有更高的优先级。因此，如果备份包含 20 个数据文件的数据库，若在备份完前 10 个数据文件后失败，则下一次就从还没有备份的那 10 个数据文件开始备份。

18.2.2　调整 MML 层

通过前面章节的学习，可知 MML 是 Oracle 提供的一种 API 接口，该接口为各类提供外部备份解决方案的供应商提供软件接口（例如磁带设备）。每个 RMAN 组件都可能

需要一些调整，其中也包括 MML 层。对于 MML 备份设备来说，需要考虑很多问题。许多 MML 备份设备是在异步模式中运行的，但是如果 MML 备份设备不在异步模式中运行，就会带来很大的问题。另外，DBA 偶尔会在为备份分配信道时设置 rate 参数，由于这会人为地导致性能瓶颈的问题，所以我们通常不会执行这样的操作。

此外，一些 MML 供应商提供了多种可配置的参数(如可配置的缓冲区大小)，我们可以在供应商指定的参数文件中配置这些参数，这些参数文件为用户提供了各种调整可能性。

与 MML 层相关的其他因素有：所用备份设备的传输速率、压缩率、数据流和数据块大小，我们必须仔细分析这些因素，以便更好地调整 RMAN 备份的性能。

18.2.3 确认与数据库相关的 RMAN 问题

从很多方面来看，RMAN 调整与 SQL 调整近似。RMAN 使用 Oracle 数据库与使用 SQL 语句的情况类似。例如使用缓冲区缓存，执行动态 SQL 调用，调用存储 PL/SQL 包等。这些操作，如定时等待事件，将会在 Oracle 数据库创建的统计信息中显示。通过利用几个视图可以帮助您在遭遇问题时分清问题的种类以及问题产生的来源。

RMAN 性能调整可以使用许多的视图。本书不是关于性能调整的书籍，因此主要从 RMAN 有关的方面提供一些知识。有关 RMAN 调整可以参考的视图包括：

- V$RMAN_BACKUP_JOB_DETAILS 视图
- V$ACTIVE_SESSION_HISTORY 视图
- V$SESSION 视图
- V$PROCESS 视图
- V$SESSION_LONGOPS 视图
- V$BACKUP_ASYNC_IO 视图
- V$BACKUP_SYNC_IO 视图

性能产生的问题存在大量的潜在根源。在使用 RMAN 执行读取操作时，如读取控制文件，以下的组件都可能出现性能问题：

控制文件

RMAN 常常需要读取控制文件以获取 RMAN 元数据。若控制文件的 I/O 速度缓慢(也许由于缓慢的磁盘响应时间)，将会降低 RMAN 操作的速度。过去，一些与 RMAN 有关的错误引起了与数据库控制文件有关的性能问题，因此如果您经历过与控制文件有关的无法解释的性能问题，应该让 Oracle 参与解决该类问题。

恢复目录

如果使用了恢复目录，RMAN 将频繁访问该目录以获取 RMAN 元数据。若恢复目录处于缓慢的 I/O 操作中，则也将会降低 RMAN 操作的速度。通常恢复目录与备份的数据库为不同的数据库。因此，性能问题的产生可能来自于恢复目录数据库，也可能来自备份的数据库。当对 RMAN 有关的问题进行性能调整时，需要观察备份数据库和恢复目录数据库的统计数据。过去，一些与 RMAN 有关的错误引起了与恢复目录有关的性能问

题，因此如果您经历过与控制文件有关的无法解释的性能问题，应该让 Oracle 参与解决该类问题。

读取内存缓冲区

与其他所有数据库操作类似，内存极其重要。需要适当地配置系统全局区(SGA)。通常，目标数据库上的内存问题是引起 RMAN 问题的基本原因之一。恢复目录数据库的内存问题将导致产生严重的性能问题。

读取数据块

RMAN 会从内存或磁盘读取数据块。如果磁盘不能充分地响应，I/O 速率将会出现问题。这种情况将会影响 RMAN 操作性能。同样地，如果系统全局区太小，则会增加从磁盘读取数据块的频率，这也会对 RMAN 的性能造成影响。

写入操作(例如数据文件还原)也会导致性能问题。包含在读取操作的问题多数也会出现在写入操作中。与写入操作的性能问题有关的数据库组件如下：

- 控制文件
- 恢复目录
- 磁带/磁盘写入操作

与从磁带或磁盘读取一样，可获得的 I/O 速率将对性能造成影响。

- 写入内存缓冲区

本节将讨论几个视图，可利用这些视图监视和调整 RMAN 性能。使用这些视图可以了解数据库、RMAN、MML 的运行情况。也可以使用这些视图确定备份或者恢复进程执行的时间，以及还需要多长时间完成备份及恢复操作。下面讨论这些视图以及如何最大程度上地使用它们：

1. V$RMAN_BACKUP_JOB_DETAILS 视图

V$RMAN_BACKUP_JOB_DETAILS 视图对数据库上的每个备份提供了信息。该视图提供了开始时间，停止时间，过去时间以及与 RMAN 关联的备份操作的字节等信息。以下是一个关于该视图的示例：

```
select /*+ RULE */ session_key, session_recid,
start_time, end_time, output_bytes, elapsed_seconds
from v$rman_backup_job_details
where start_time >= sysdate-180
and status='COMPLETED'
And input_type='DB FULL';

SESSION_KEY SESSION_RECID START_TIM END_TIME  OUTPUT_BYTES ELAPSED_SECONDS
----------- ------------ --------- --------- ------------ ---------------
        456          456 11-JAN-09 11-JAN-09    228353024             397
        461          461 12-JAN-09 12-JAN-09    229755904             422
```

在该例中，可以看到成功执行了两个备份操作。通过分析时间，可以获取备份花费的时间，以及备份时间是否在增长等趋势性信息。通过观察备份执行时间的趋势，可以在实际产生问题之前解决问题。

2. V$SESSION_LONGOPS 与 V$SESSION 视图

V$SESSION_LONGOPS 视图在备份或还原操作执行期间可用于确定操作时间开销情况，以及预期完成时间情况。将该视图与 V$SESSION 视图连接，可以获得关于 RMAN 备份或恢复会话的其他信息。以下是一个在数据库备份期间连接 V$SESSION_LONGOPS 视图和 V$SESSION 视图的示例：

```
SQL> select a.sid, a.serial#, a.program, b.opname, b.time_remaining
  from v$session a, v$session_longops b
  where a.sid=b.sid
  and a.serial#=b.serial#
  and a.program like '%RMAN%'
  and time_remaining > 0;
   SID SERIAL# PROGRAM   USERNAME  OPNAME                    TIME_REMAINING
  ---- ------- -------- --------- -------------------- ------------------
    10       8 RMAN.EXE SYS RMAN: aggregate input                    1438
    14       3 RMAN.EXE SYS RMAN: full datafile backup               7390
```

在上例中，可观察到一个 RMAN 进程正在执行备份。其连接到 SID 为 14 的数据库。剩余的时间为 7390 秒，是预期的备份执行时间。通过该报告，可以清楚地知道备份需要花费的时间。通过与 V$SESSION 视图的连接还获得了关于 RMAN 会话的其他信息，包括用户名和程序名等。

3. V$ACTIVE_SESSION_HISTORY 视图

Oracle 数据库提供了一个称为活动会话历史(Active Session History，ASH)的特性，该特性可提供关于 Oracle 数据库操作的与会话有关的历史信息。ASH 是一个可用于查看历史运行信息的强有力工具，提供的信息可用于有效地开展 RMAN 的性能调整操作。使用 ASH 可用于查看各个会话经历的等待时间以及包含哪些相关联的等待。以下是一个查询示例：

```
-- First we use V$SESSION to get the session
-- specific information if possible.
select sid, serial#, program
  from v$session
  where lower(program) like '%rman%';

    SID    SERIAL# PROGRAM
------ ---------- ----------
    125        149 rman.exe
    128        130 rman.exe
    134        164 rman.exe
```

```
-- Note that runtime session information may not always be available.
Set lines 132
Column session_id format 999 heading "SESS|ID"
Column session_serial# format 999 heading "SESS|SER|#"
Column event format a40
Column total_waits format 9,999,999,999 heading "TOTAL|TIME|WAITED|MICRO"
Select session_id, session_serial#, Event, sum(time_waited) total_waits
From v$active_session_history
Where sample_time > sysdate - 1
-- The next line can be remarked out if you don't know
-- the session information.
And session_id||session_serial# in (120102, 128102, 134129)
And program like '%rman%'
And session_state='WAITING' And time_waited > 0
Group by session_id, session_serial#, Event
Order by session_id, session_serial#, total_waits desc;
```

SESS ID	SESS SER #	EVENT	TOTAL TIME WAITED MICRO
125	149	control file single write	1,388,961
125	149	control file sequential read	45,964
125	149	control file parallel write	3,789
128	130	RMAN backup & recovery I/O	192,263,005
128	130	control file single write	1,095,253
128	130	control file parallel write	529,012

4. V$BACKUP_ASYNC_IO 与 V$BACKUP_SYNC_IO 视图

V$BACKUP_ASYNC_IO 与 V$BACKUP_SYNC_IO 视图包含 RMAN 异步和同步备份操作的详细信息。这些视图实质上是暂时的，因此在数据库关闭后将被清除。视图中包含有每个同步或异步备份或恢复操作的行信息。其中最有用的信息可能是 TYPE 列被设置为 AGGREGATE 的记录中的 EFFECTIVE_BYTES_PER_SECOND 列，该列表示备份或者恢复的对象每秒执行的字节数。该数字应该与列出的备份硬件的读/写率近似。如果 EFFECTIVE_BYTES_PER_SECOND 列中的值比备份硬件读/写速率值低很多，则应当查找备份进程中出现的问题。问题可能由多种情况产生，可能是 CPU 超载，也可能是网络饱和，也可能是 MML 接口的配置问题，或者供应商备份解决方案等问题。

注意：
若在 V$BACKUP_SYNC_IO 视图中看到数据，则表明未执行异步备份操作。在此情况下，应该查找执行同步备份操作的原因。

以下是一个利用 V$BACKUP_ASYNC_IO 视图查询的示例以及数据库备份操作完成后的结果。

```
select device_type "Device", type, filename,
to_char(open_time, 'mm/dd/yyyy hh24:mi:ss') open,
to_char(close_time, 'mm/dd/yyyy hh24:mi:ss') close,
elapsed_time ET, effective_bytes_per_second EPS
from v$backup_async_io
where close_time > sysdate - 30
order by close_time desc;

Device              TYPE
------------------ ---------
FILENAME
--------------------------------------------------------------------------------
OPEN                 CLOSE                          ET         EPS
-------------------- -------------------- ---------- ----------
DISK                INPUT
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_10
/o1_mf_nnndf_TAG20090910T110421_5blddpg9_.bkp
09/10/2009 11:15:35 09/10/2009 11:16:38       6300    4078836

DISK                OUTPUT
/ora01/oracle/rob1/rob1/system01.dbf
09/10/2009 11:15:35 09/10/2009 11:16:37       6200    11838761

DISK                AGGREGATE
09/10/2009 11:15:35 09/10/2009 11:16:37       6200    21986271

DISK                OUTPUT
/ora01/oracle/rob1/rob1/sysaux01.dbf
09/10/2009 11:15:36 09/10/2009 11:16:24       4800    13107200

DISK                AGGREGATE
09/10/2009 11:10:54 09/10/2009 11:11:06       1200    42269355

DISK                OUTPUT
/ora01/oracle/rob1/rob1/sysaux01.dbf
09/10/2009 11:10:55 09/10/2009 11:11:06       1100    9245789

DISK                INPUT
/oracle/app/oracle/flash_recovery_area/ROB1/backupset/2009_09_10
/o1_mf_nnndf_TAG20090910T110421_5blddpg9_.bkp
09/10/2009 11:10:54 09/10/2009 11:11:05       1100    1429132

DISK                OUTPUT
/ora01/oracle/rob1/rob1/system01.dbf
```

```
09/10/2009 11:10:54 09/10/2009 11:11:05        1100    9341114

DISK                OUTPUT
/ora01/oracle/rob1/rob1/undotbs01.dbf
09/10/2009 11:10:55 09/10/2009 11:11:00         500    58720256

DISK                OUTPUT
/ora01/oracle/rob1/rob1/users01.dbf
09/10/2009 11:10:55 09/10/2009 11:10:56         100    9175040

10 rows selected.
```

在该示例中，可以看到通过 RMAN 执行从数据库传送到备份集的有效速率。另外也可以看到备份的数据文件的名称，以及备份实际的启动时间，结束时间等。

另外一个衡量备份进程效率的方法是使用 V$BACKUP_ASYNC_IO 视图。该视图包含几个有用的列，如表 18-1 所示。

表 18-1　V$BACKUP_ASYNC_IO 视图中有用的列

列　名	描　述
IO_COUNT	I/O 计数总数
READY	缓冲区立即可用的异步 I/O 调用数
SHORT_WAITS	请求一个缓冲区但缓冲区不能立即使用，而在以非阻塞方式轮询 I/O 完成之后变为可用的次数
LONG_WAITS	缓冲被请求时不可用，Oracle 必须等待 I/O 设备的次数

为确定是否是由于 I/O 引起的问题，可以查看 I/O 和长等待的比率(LONG_WAITS/IO_COUNTS 列)。如下代码所示：

```
select b.io_count, b.ready, b.short_waits, b.long_waits,
    b.long_waits/b.io_count, b.filename
    from v$backup_async_io b;
    IO_COUNT READY SHORT_WAITS LONG_WAITS B.LONG_WAITS/B.IO_COUNT
    ---------- ---------- ---------- ---------- ----------------------
    FILENAME
    ----------------------------------------
             2          1          0          1                    .5
    D:\ORACLE\ADMIN\RECOVER\ARCH\ARC00052.001
             2          1          0          1                    .5
    D:\ORACLE\ADMIN\RECOVER\ARCH\ARC00046.001
             2          1          0          1                    .5
    D:\ORACLE\ADMIN\RECOVER\ARCH\ARC00051.001
             2          1          0          1                    .5
    D:\ORACLE\ADMIN\RECOVER\ARCH\ARC00050.001
           171        107         12         52                    .304093567
```

```
D:\ORACLE\ORADATA\RECOVER\SYSTEM01.DBF
         11            8             2          1                    .090909091
D:\ORACLE\ORADATA\RECOVER\RECOVER_UNDOTBS_01.DBF
          6            4             0          2                    .333333333
D:\ORACLE\ORADATA\RECOVER\TOOLS01.DBF
          6            3             0          3                    .5
D:\ORACLE\ORADATA\RECOVER\USERS01.DBF
          6            4             1          1                    .166666667
D:\ORACLE\ORADATA\RECOVER\RECOVER_TESTRBS_01.DBF
          3            1             0          2                    .666666667
D:\ORACLE\ORADATA\RECOVER\INDX01.DBF
          2            1             0          1                    .5
D:\ORACLE\ORADATA\RECOVER\TOOLS02.DBF
```

该查询返回的数字清楚地指示是由于某种 I/O 瓶颈导致的问题(在此环境中，是由于单一 CPU 负担过重引起的问题)。

18.3 跟踪 RMAN 会话

有时仅使用视图难以追踪问题。需要着手研究根源。这意味着需要跟踪与 RMAN 操作有关的 Oracle 会话。由于 RMAN 通常为完成其工作建立多个 Oracle 会话，使得跟踪工作比较复杂。本节将介绍跟踪的有关概念，以及如何开始跟踪工作。跟踪，与调整类似，是一个自行其事的话题。如果发现需要跟踪 RMAN 的会话，一般可能都遇到严重的问题。如果处于这些情况中，就需要对 Oracle 跟踪做更加细致的研究并与 Oracle 公司探讨。

根据 RMAN 问题的情况，可能需要跟踪一个或所有的会话。可以使用几种方法开展对 RMAN 会话的跟踪工作。我们从简单的方法着手，然后再考虑复杂的情况。

通过启用 Oracle 事件可以完成跟踪 Oracle 的工作。Oracle 事件是 DBA 或者开发者为使 Oracle 做其通常不会做的工作所做的“设置”。每个事件都被编号，在本书环境下，我们感兴趣的是 10046 事件。在设置 10046 事件后，实际上是告知 Oracle 数据库，需要它为您创建一个跟踪文件并开始跟踪。设置跟踪的工作可以在数据库或者会话级别启用。由于跟踪会带来开销，所以需要谨慎对待跟踪，只跟踪那些需要跟踪的会话。

对跟踪文件输出的读取不是本书范围要讨论的问题。很少有对此方面的需求。然而，可能需要研究这方面的问题，或者 Oracle 需要跟踪 RMAN 会话。我们将告诉您如何执行该操作。根据需要的不同，可以使用以下几种方法跟踪 RMAN 会话：

- 从 RMAN 命令提示符下使用 SQL 命令，启用 10046 跟踪，如下所示。此时只能开始跟踪 RMAN 客户端连接到的会话。其结果是，一些会话不能被跟踪。该方法通常是跟踪会话的最简单的方法，并可能会带来良好的效果：

```
Sql "alter session set event' '10046 trace name context forever, level 12' ' ";
RMAN>backup as compressed backupset database plus archivelog delete input;
```

- 可以对恢复目录数据库开始跟踪。可以跟踪整个数据库，或者使用后面示例显示的登录触发器。以下是对恢复目录数据库进行跟踪的示例。该示例跟踪整个数据库。如果希望执行更多的技巧性操作，则可以使用下一个方法介绍的触发器代码跟踪 RMAN 连接。

```
ALTER SESSION SET EVENTS='10046 trace name context off';
```

- 若第 1 个会话的跟踪文件不能帮助确定需要处理的问题，则可以建立一个登录触发器，用该触发器在会话注册时开始跟踪工作。以下示例的触发器将跟踪连接到数据库上的所有 RMAN 会话。

```
Create or replace trigger tr_rman_logon
After logon on database
Declare
     v_program varchar2(1000);
Begin
    SELECT distinct program into v_program
    FROM v$session
    WHERE audsid = userenv('sessionid')
    and program like '%rman.exe%' and rownum < 2;
    if v_program='rman.exe'
    then
    execute immediate 'alter session set events ''10046 trace name context
    forever, level 12''';
    end if;
Exception
      when NO_DATA_FOUND then
           NULL;
End;
/
```

在 Oracle Database 11g 中，跟踪文件结果可以在自动诊断目标(Automatic Diagnostic Destination)目录结构中找到，该目录通过 DIAGNOSTIC_DEST 参数指定。可使用 11g 中的 show 参数找到 DIAGNOSTIC_DEST 参数的设置，也可以通过查看 USER_DUMP_DEST 参数的默认设置获得该目录(在 11g 中不主张使用该参数，但它仍然是一种在 ADR 中找到各种文件路径的方便办法)。在 11g 以前的各种版本中，可以通过 USER_DUMP_DEST 参数指向的目录路径找到跟踪文件的结果。

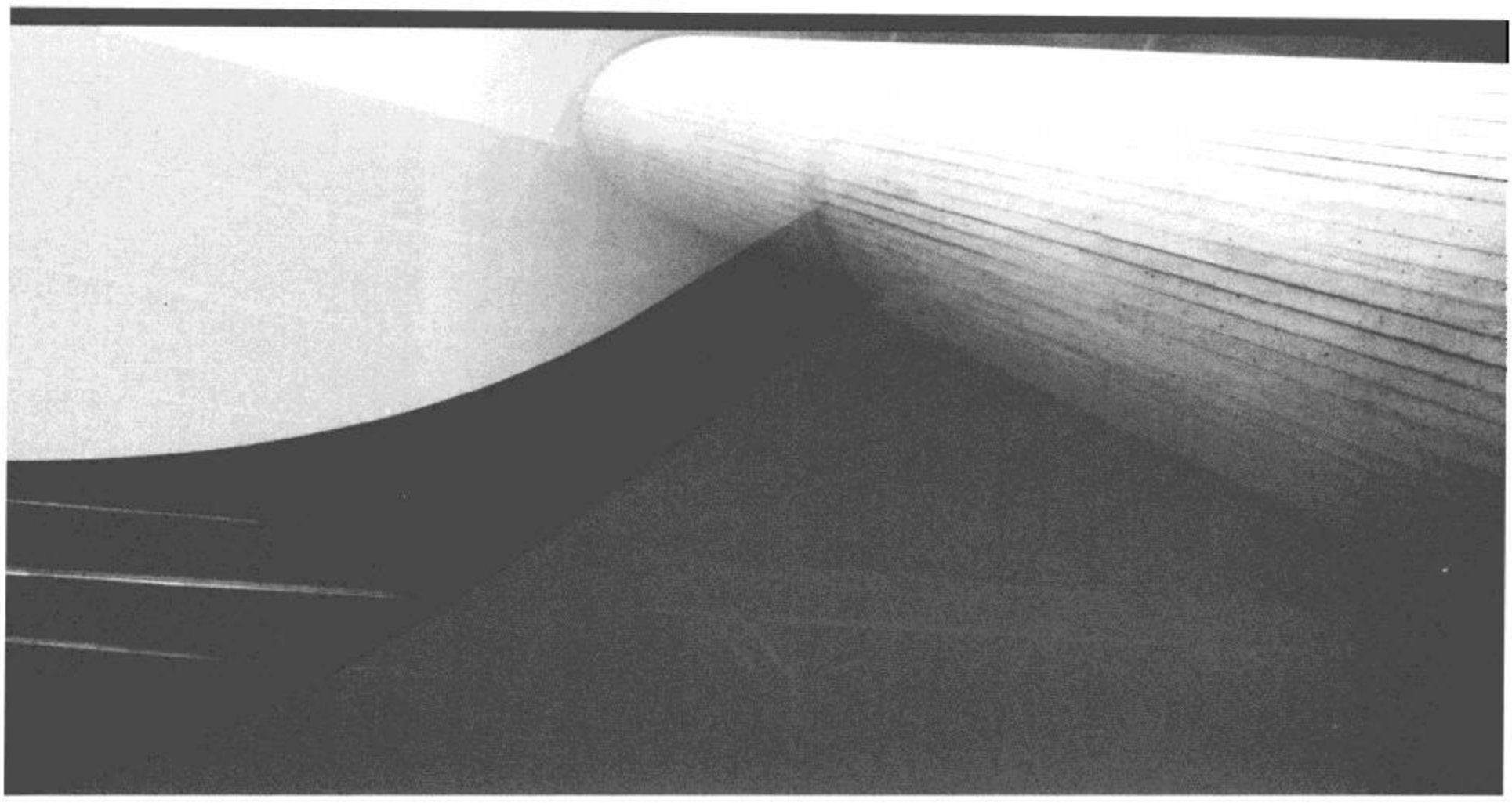
若能够使用 V$SESSION 和 V$PROCESS 视图标识会话的 ID，也可以获得指定跟踪文件的位置和名称。这些视图中都有一个称为 TRACEFILE 的列，该列会给出跟踪文件的名称和位置。

本书的脚本创建的 10046 跟踪文件非常庞大和复杂，超出了本书的讨论范围。多数的 DBA 意识到可以利用 Oracle 提供的 tkprof 工具来重新格式化这些跟踪文件。多数 DBA 也意识到如何读取及解释结果输出。如果需要更多的关于读取跟踪文件的信息，可以参考有关该问题的相关信息资源，这些资源包括书，网站以及 Oracle Metalink。

18.4 小结

只要有正确基础架构的支持，RMAN 就可以支持非常快速的备份和恢复方案。RMAN 不能良好工作的原因通常不在于数据库或 RMAN 本身，而在于底层的网络或备份设备的无效数目。

我们也讨论了 RMAN 运行过快、占用过多 CPU 或网络阻塞的情况。我们讨论了 duration 参数和它的各种选项，通过这些选项可以减少对 RMAN 操作的整体运行时间的影响。



第Ⅳ部分

Oracle 生态系统中的 RMAN

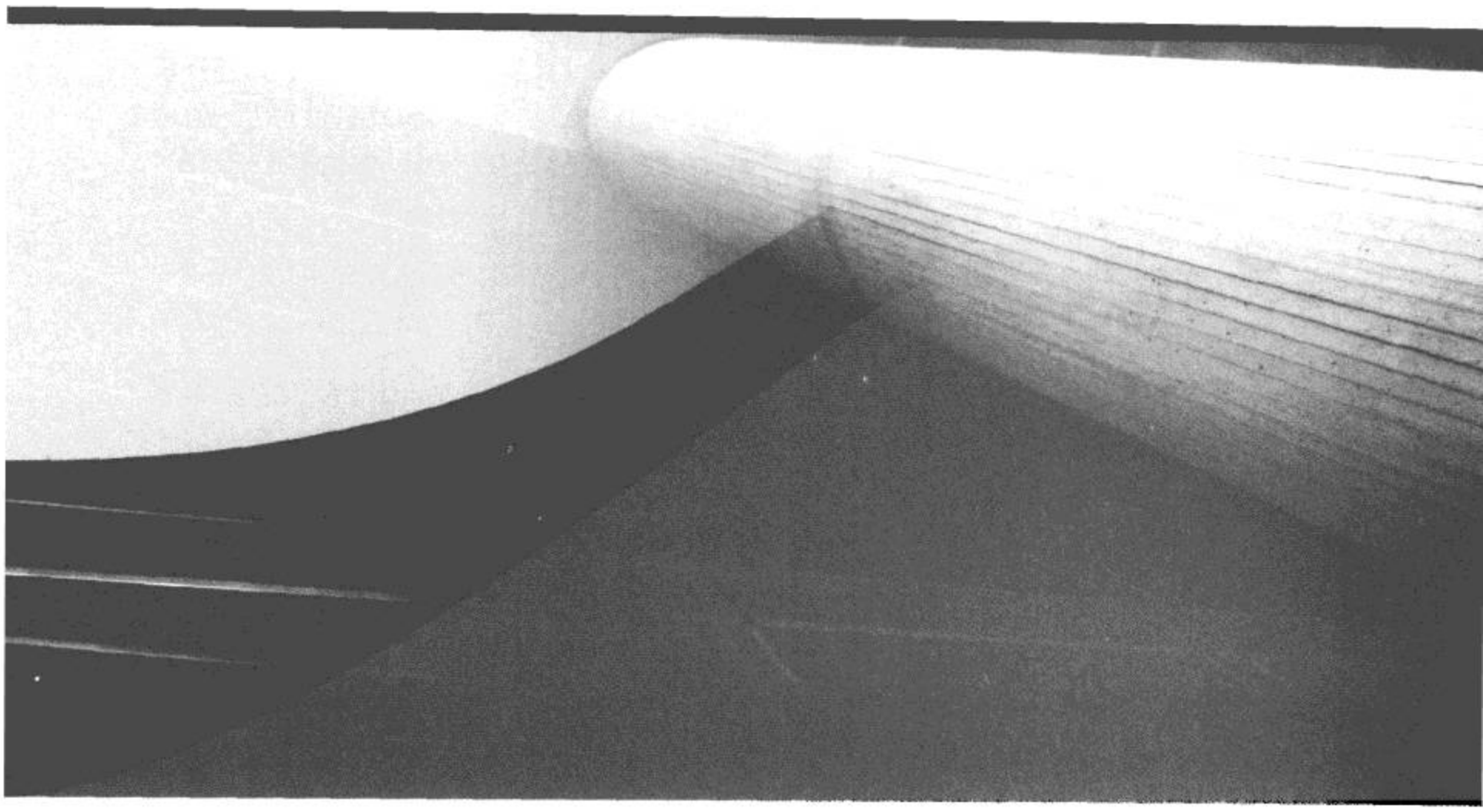
第 19 章　复制目标数据库

第 20 章　RMAN 与 Data Guard

第 21 章　RMAN 与 Real Application Clusters

第 22 章　同步和拆分技术中的 RMAN

第 23 章　RMAN 案例研究



第19章

复制目标数据库

在本书前面的介绍中已经提及本章的内容：RMAN 帮助您将备份用于灾难恢复之外的情况。实际上，我们已经讨论了使用 RMAN 进行损坏检查和跨平台的数据库转换(参见第 14 章)。备份的另一种常用方法是用于建立测试和开发所需的产品数据库副本。在数据库领域中，我们将这种方法称为复制。

数据库复制为测试版本升级、新应用程序的初次发布、甚至是故障补丁提供了很好的方法。也可将它用于准备与事务处理数据库分离的报告数据库。数据库备份在这些方面都有其作用，它们可以用于将数据库还原到另一个系统，或者还原到相同系统上的另一个磁盘。备份还可以执行针对性能的负载测试，或者尝试新的硬件配置。通过使用复制数据库，我们几乎能够完全了解改变产品环境会出现的情况。使用 RMAN 可以创建备用数据库，备用数据库则保存了用于灾难恢复的最新数据。

RMAN 通过 duplicate 命令帮助用户复制数据库。duplicate 命令把复杂的东西隐藏起来，您有必要先了解该命令的用法。此外，为了顺利地执行复制操作，还必须完成大量的预备工作。不过，一旦了解这些内容并完成一些练习之后，您会发现数据库复制是 RMAN 备份和恢复中一个非常实用的程序。或许您已经了解了这一点，但是利用 RMAN 的接口可节省大量编写脚本的时间。

19.1 RMAN 复制的基础

RMAN 的 duplicate 命令非常简单，它把高度的复杂性隐藏起来。如果具有将数据库的映像副本备份还原到其他系统中的经历，您就会知道必须跟踪大量的信息，这些信息包括文件名、文件位置、备份位置、归档日志信息、ftp 进程、移动磁带等。太多的信息需要跟踪，但如果使用 RMAN，则可以直接执行下面的命令：

```
duplicate target database to aux1;
```

这条命令提示 RMAN 执行将目标数据库复制到另一个数据库的完整进程。当然，必须首先完成一些准备工作，但是一旦熟悉了复制进程的体系结构，您会发现准备工作所耗费的精力与巨大的回报相比是微不足道的。

19.1.1 使用 RMAN 复制的原因

为什么需要复制？难道就不能只将控制文件复制到新位置就开始执行还原和恢复操作吗？答案是：可以。没有理由阻止您这么做。唯一存在的问题是，您应该获得一个准确的副本——不仅仅是您想要的数据库中的数据(您希望是相同的)，而且应该包括数据库的元数据(可能令人头疼)。在使用了恢复目录的情况下，如果不使用 RMAN 复制来复制数据库，就存在许多问题。由于 RMAN 基于数据库 ID(DBID)在恢复目录中注册数据库，因此即使存在同名的数据库(init.ora 中的 DB_NAME 参数表示数据库名)，数据库 ID 也能唯一地标识数据库。如果不使用 duplicate 命令，就会存在具有相同 DBID 的两个数据库。此时，如果尝试在相同的目录中注册复制的数据库，就会得到下面的错误消息：

```
rman-20002: target database already registered in recovery catalog
```

这条错误消息很容易让人误解，它表明您还没有注册数据库！如果您不理会，并继续执行数据库备份操作，就会得到一条更奇怪的错误消息：

```
rman-20011: target database incarnation is not current in recovery
    catalog
```

您会更加困惑，然而您继续执行 reset database 命令，并认为这次可以备份数据库，但是这样却会导致更大的问题。此时，您也许会连接 RMAN 与原始的产品服务器，并且试图执行备份操作，但是又会得到一条相同的错误消息：

```
rman-20011: target database incarnation is not current in recovery
    catalog
```

产生这条错误消息的原因是 RMAN 认为用户复制的只是产品系统的还原版本，所以将该复制当作产品服务器的当前对应物，这样 RMAN 就完全不知道真正的产品服务器是什么。虽然我们能够将对应物重置回与产品服务器匹配的一个对应物，但是实际上已经破坏了恢复目录，所以应当取消数据库的注册，然后在恢复目录中重新注册这个数据库(参阅第 3 章)。当然应该注意到，每个 Oracle 发布版本的文档信息中都清楚地指明不应该将 RMAN 连接到一个测试数据库和恢复目录，就是为了防止该问题的发生。

如果没有使用恢复目录，非完全拷贝的复制也会造成严重的损害。例如，在与主数据库相同的系统中复制数据库，然后使用 RMAN 在磁带上备份这两个数据库，由于没有使用恢复目录，所以您会打开这两个实例的控制文件自动备份。某天，您突然丢失了一个磁盘阵列信息，整个系统出现故障，同时还丢失了所有数据库和控制文件。此时，您会认为由于自动备份了控制文件，所以只需要使用其中一个备份就能够恢复系统。

然而，事实并非如此。restore controlfile from autobackup 命令使用 DB_ID 跟踪控制文件的自动备份，由于两个数据库都被备份在同一个磁带上，所以该命令可能从错误的数据库备份中还原控制文件，用户得到的就是含有错误数据的错误文件。

第 3 个原因，当您在使用 Oracle Enterprise Manager Grid Control 环境时将会出现。Grid Control 对其管理的所有目标都需要具有唯一性。对一个数据库执行复制，就会导致两个数据库具有相同的元数据。当试图利用 Grid Control 对两个数据库进行管理时，就会给 Grid Control 的管理带来麻烦。它要么会拒绝发现数据库，要么让您反复验证新的复制如何与监听器一起注册(注意：即使将复制数据库放置于不同的服务器上，这种情况仍然会出现)。

显然，这几种情况都能够被修正，并且不会丢失数据，但是这样会带来时间上的损失，它们可能会超过用户允许的停机时间。除了针对这些问题之外，在 RMAN 中使用复制能简单地提供一个 RMAN 接口以使得操作尽可能地简单，这是非常明智的选择。

19.1.2　RMAN 复制的不同类型

Oracle 11*g* 提供了多种使用 RMAN 实现数据库复制的方法。过去，在开始复制前需要做一些重要的准备工作：访问要复制的目标数据库以及访问已存在的备份。在新的环境中，充分考虑了使用 RMAN 规划和执行复制操作的灵活性，这些准备工作都可以省略。

1. 使用 RMAN 备份复制并访问目标数据库

该方法是从 9*i* 到 10*g* 的传统的 RMAN 复制方法：利用已有的 RMAN 备份，在新位置为已有的目标数据库创建一个副本。为理解复制到什么程度，复制什么，RMAN 在复制操作期间始终与已有数据库的控制文件保持连接。

2. 使用 RMAN 备份复制并无需访问目标数据库

11*g* 的新方法，当目标数据库不可用时仍然可以执行复制操作。当目标数据库位于不同的对应物中，处于无法访问的网络位置，或者复制操作不可用时，该方法能够发挥作用。在此情况下，RMAN 能够从恢复目录中获得需要的额外的参数，这些参数需要传递

给 duplicate 命令。在本章的 19.5 节中将讨论该问题。

3. 不使用备份复制活动的数据库

复制活动的数据库听起来具有新意，实际上在 10g 版本的企业管理器中该操作已经有所体现。在 10g 版本中，企业管理器可以通过直接调用 dbms_backup_restore 包来复制数据库，而不是使用 RMAN 客户端命令语法，可以实现实时复制数据库的数据文件副本，然后将数据文件移动到新位置。

在 11g 中，上述功能整合到 RMAN 命令语法中，可以实现同样的功能。不用依赖已有的 RMAN 备份，RMAN 就可以连接到目标数据库，直接执行针对数据库的实时备份操作，并将文件写入到指定的位置。除了调用操作的语法有些差异外，没有什么特别的不同。

以上讨论的三种复制类型依赖相同的基础体系结构。我们将讨论用于基于备份的，目标数据库-连接的复制操作以理解其基本的概念。

19.1.3 复制体系结构

本节将介绍复制的工作原理。RMAN 连接到目标数据库或恢复目录(如果使用了恢复目录)，用户通过这个连接才能够访问目标数据库控制文件中的备份位置的详细信息。连接目标数据库或目录后，还必须连接辅助实例。复制的数据库将驻留在辅助实例中。启动复制操作前，必须为辅助实例构建一个 init.ora 文件，同时以 NOMOUNT 模式启动辅助实例，这样就可以初始化内存段，并且 RMAN 可以生成到这个辅助实例的 SYSDBA 连接。由于辅助实例仍然不包含控制文件(复制操作会负责创建)，因此我们不可能加载这个辅助实例。

生成上述连接后，就可以执行简单的 duplicate 命令，如下所示：

```
duplicate target database to aux1;
```

或者由于辅助实例所处的位置，duplicate 命令也可以非常复杂，如下所示：

```
run {
set until time = '08-JUL-2002:16:30:00';
duplicate target database to aux1 pfile=/u02/oracle/admin/aux1/pfile/
init.ora
nofilenamecheck
device type sbt parms "env=(nb_ora_serv=rmsrv)"
logfile
'/u04/oracle/oradata/aux1/redo01.log' size 100m,
'/u05/oracle/oradata/aux1/redo02.log' size 100m,
'/u06/oracle/oradata/aux1/redo03.log' size 100m;}
```

复制进程可以分为以下这些不同的阶段：

(1) RMAN 确定备份的属性和位置。

(2) RMAN 在辅助实例上分配辅助信道。

(3) RMAN 将数据文件还原到辅助实例中。

(4) RMAN 构建新的辅助控制文件。

(5) 如果有必要，RMAN 从备份中还原归档日志，同时执行任何必要的恢复操作。

(6) RMAN 重置辅助实例的 DBID，并且使用 open resetlogs 命令打开辅助数据库。

首先，RMAN 设置运行时(runtime)参数，如 duplicate 命令中的 until time 子句。然后，RMAN 基于这些参数检查目标数据库的控制文件或恢复目录，确定适当的备份。随后，RMAN 构建访问备份(包括如何访问及访问哪些备份)的远程过程调用，但是并不会在目标数据库上执行代码，反之，RMAN 在辅助实例上创建一个称为辅助信道(auxiliary channel)的信道进程，RMAN 向这个信道传递对 DBMS_BACKUP_RESTORE 数据包的调用。最后，辅助实例访问备份和还原所有必需的数据文件。图 19-1 示例说明了磁盘备份和磁带备份的复制操作。

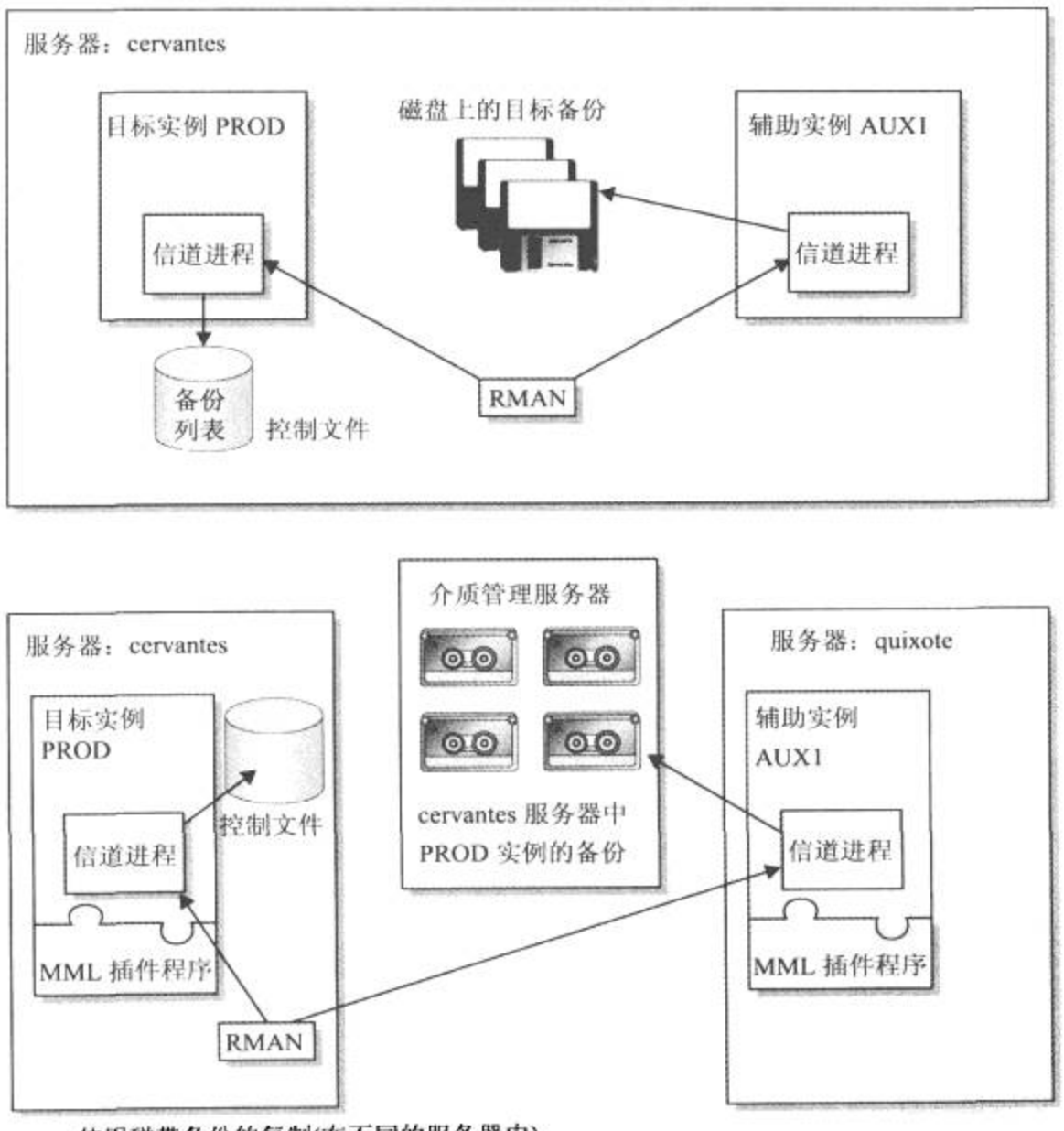


图 19-1　复制操作概略图

1. 辅助信道配置

为了执行复制操作，RMAN 必须在辅助实例上分配一个或多个信道进程。从 Oracle 9*i* 开始，由于使用目标实例控制文件中存储的永久配置参数可以自动创建辅助信道，所以

不必在复制时手动地分配辅助信道。辅助信道的构成主要来自于为目标信道创建的参数，其默认设备类型和并行化级别都使用与目标信道设置相同的永久参数。因此，如果使用磁盘上的备份执行复制操作，就不需要再配置辅助信道。然而，如果使用磁带上的备份执行复制操作，就必须配置辅助信道，以包含目标信道包含的全部介质管理器环境参数。例如，下面的代码将默认设备类型设置为磁带，将默认的并行化级别设置为 2，然后使用正确的参数配置两个辅助信道：

```
configure default device type to sbt;
configure device type sbt parallelism 2;
configure auxiliary channel 1 device type sbt parms
= "env=(nb_ora_serv=mgtserv, nb_ora_class=oracle)";
configure auxiliary channel 2 device type sbt parms
= "env=(nb_ora_serv=mgtserv, nb_ora_class=oracle)";
```

2. 将数据文件还原到不同的文件位置

加载新的控制文件后，RMAN 就可以开始还原数据文件。如果在驻留目标数据库的相同服务器上复制数据库，就必须修改数据文件还原的位置。

即使将数据文件还原到不同的服务器上，加载点和目录结构中的差异也要求使用新的文件位置。我们可以使用 3 种方法来修改复制进程的数据文件还原步骤以指向新的文件位置。

第一种方法是使用下面所示的 configure 命令，为需要配置新位置的所有数据文件配置 auxname，这个配置存储在目标数据库的控制文件中：

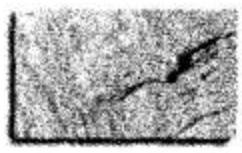
```
configure auxname for datafile 1 to '/u04/oradata/aux1/system01.dbf';
configure auxname for datafile 2 to '/u04/oradata/aux1/undo01.dbf';
...
```

第二种方法是在 run 命令中指定新的数据文件名，与上一版本相同，如下所示：

```
run {allocate channel c1 type 'sbt_tape';
set newname for datafile 1 to '/u04/oradata/aux1/system01.dbf';
set newname for datafile 2 to '/u04/oradata/aux1/undo01.dbf';
duplicate target database to aux1;}
```

最后一种方法是使用辅助数据库的 init.ora 文件中的 DB_FILE_NAME_CONVERT 参数为数据文件设置新的位置，我们需要向这个参数传递两个字符串：第一个字符串是数据文件在目标数据库上的原始位置，第二个字符串是数据文件在辅助实例中的新位置。使用如下所示的匹配文件对可以完成上述操作：

```
db_file_name_convert=(
    '/u02/oradata/prod/system01.dbf', '/u02/oradata/aux1/system01.dbf',
    '/u03/oradata/prod/prd_data_01.dbf',
'/u03/oradata/aux1/prd_data_01.dbf')
```



注意：
在 11gR2 版本中，为了使脚本构建更加简单，对 SET NEWNAME 功能进行了显著的改善。可以对整个表空间，或者整个数据库执行 SET NEWNAME 操作。例如：

```
SET NEWNAME for tablespace USERS01 to '/u01/%U';
```

上述命令将所有数据文件重命名到新位置，由变量%U 定义使用唯一性命名方法。这种方式简化了当新位置有一个不同的，但一致的文件位置时的复制脚本的编写工作。
另外，存在一条 Metalink 注意事项，用于说明使用 Oracle 支持的访问，以提供动态 SQL 访问为用户管理文件生成 SET NEWNAME 命令的方法，或者将文件转换到 Oracle 的自动存储管理器(ASM)和 Oracle 的自动管理文件(OMF)基础架构。该注意事项为 Note 549972.1，“RMAN: SET NEWNAME Command SQL”。

DB_FILE_NAME_CONVERT 是一个简单的字符串转换参数，因此我们能够简单地传递一个要修改的目录名。例如，数据文件分布在 4 个加载点，但是它们的目录结构中都包含 prod，所以从 V$DATAFILE 视图中 select 的结果如下所示：

```
Select name from v$datafile;
------------------------------------------------------------
/u02/oradata/prod/system01.dbf
/u03/oradata/prod/prd_data_01.dbf
/u04/oradata/prod/indx_prd01.dbf
/u05/oradata/prod/temp01.dbf
```

此时不必构建每个匹配文件对，可以简单地执行下面的操作：

```
db_file_name_convert=('prod' , 'aux1')
```

只要与辅助数据库的文件位置有关的其他信息(如加载点)都是相同的，就能够使用这条命令来修改数据文件的位置。

3. 创建新的控制文件

还原所有文件后，复制操作会为辅助实例创建新的控制文件。使用 duplicate 命令中概述的各种参数，RMAN 在辅助实例上执行 create controlfile 命令。创建控制文件后，则可以加载这个辅助数据库。此时，RMAN 执行切换操作切换到新的文件，这个切换操作可以使 RMAN 修改辅助站点上的新控制文件以指向新的数据文件的位置。

4. 恢复和归档日志

还原和切换文件后，就能够对数据库执行恢复操作，此时既可以将数据库恢复到当前

时间点，也可以将数据库恢复到 until time 子句中指定的时间点。要执行恢复操作，RMAN 需要访问归档日志。如果 RMAN 备份了归档日志，RMAN 就能够将这些归档日志从备份位置还原到辅助数据库的 init.ora 文件中指定的 LOG_ARCHIVE_DEST 位置。我们也可以手动地将归档日志移动到新实例所要求的位置，这样 RMAN 就能够不需要还原归档日志而在磁盘上查找到这些文件。如果在目标数据库当前驻留的相同服务器上执行复制操作，RMAN 就能够在目标数据库的 LOG_ARCHIVE_DEST 位置中查找到归档日志。

还原归档日志后，RMAN 会执行一些必需的恢复操作。如果没有指定结束恢复操作的时间点，RMAN 会还原到最后一个可用的归档日志(在 V$ARCHIVE_LOG 视图中能够查找到该日志)，然后结束恢复操作。在复制操作期间，RMAN 不能进一步检查联机重做日志文件的恢复信息。到达归档日志的结束点后，RMAN 则停止恢复操作。如果 RMAN 从备份中还原了归档日志，这些归档日志就会在恢复操作完成后被删除。

5. 修改数据库 ID(DBID)

介质恢复完成后，数据库将处于一致的状态中，此时 RMAN 将修改新复制的 DBID。由于在这个时间点之前的所有操作都要求复制数据库与目标数据库具有相同的 DBID，所以 RMAN 必须等待复制数据库中的其他操作全部完成。如果控制文件具有不同的 DBID，复制数据库在介质恢复期间就不能应用归档日志。

修改 DBID 的进程非常简单。RMAN 有一个调用 dbms_backup_restore.zerodbid( )的进程。在数据库处于加载但未打开的状态时，这个包会查看文件头并将每个文件头中的 DBID 清零。随后，RMAN 关闭这个数据库并再次重新创建辅助控制文件。重新构建控制文件时，Oracle 会检查文件头中的 DBID，如果没有发现任何 DBID，Oracle 就会生成一个新的 DBID 并且将其加入到每个文件头中。

zerodbid 进程的注意事项

正如想象的那样，文件头中没有 DBID 就关闭数据库或重新构建控制文件会使得数据库处于非常容易受攻击的状态中，然而在 RMAN 的复制进程中，容易出错的元素受到严格的控制，因此不必过分担心会出现这种情况。我们指出这一点是因为，使用 zerodbid 进程能够使任何数据库生成新的 DBID，只需要加载数据库和运行下面的代码：

```
execute sys.dbms_backup_restore.zerodbid(0);
```

然后，使用下面的 set 参数关闭数据库和重新构建控制文件：

```
create controlfile SET database <db_name> resetlogs..
```

好了，现在就能够得到一个新的 DBID。是不是非常简单？

但是，如果在复制操作期间没有完全控制 RMAN 的环境，就会产生许多错误。例如，如果没有彻底关闭数据库，并且在使用 open resetlogs 命令前需要执行介质恢复操作，这样就会出现问题。此时，归档日志具有不同的 DBID，由于数据库处于不一致的状态中，所以用户无法打开数据库，也无法修正该错误。一个文件意外地脱机也会导致发生同样

的状况，此时这个文件在用户执行 open resetlogs 命令时不能得到新的 DBID，因此无法使该文件联机。用户会得到如下所示的错误消息：

```
ORA-01190: control file or datafile <name> is from before the last RESETLOGS
```

因此，如果要手动执行 zerodbid 进程，必须非常谨慎。除此之外，还有一种更好的方法能够用于修改数据库的 DBID。Oracle 在 9*i* R2 中介绍了一个称为 DBNEWID 的实用程序，这个实用程序不需要手动调用 DBMS_BACKUP_RESTORE 包就能够安全地为数据库生成新的 ID。我们将在本章后面的 19.5.1 小节中讨论 DBNEWID 实用程序。

6. 在辅助站点上创建日志文件

在完成复制进程并执行最后一条 open resetlogs 命令时，RMAN 必须为辅助数据库构建全新的日志文件。这种情况在执行 resetlogs 命令时通常会发生，不过在使用 duplicate 命令时需要确定新的重做日志文件的定义。如果在目标数据库所在的系统上执行复制操作，至少需要重命名日志文件。

如果要修改新的数据库使用的日志文件的大小、数目和位置，可以在执行 duplicate 命令时指定新的重做日志文件的定义，如下所示：

```
duplicate target database to aux1
    pfile=/u02/oracle/admin/aux1/init.ora
    logfile
    '/u04/oracle/oradata/aux1/redo01.log' size 100m,
    '/u05/oracle/oradata/aux1/redo02.log' size 100m,
    '/u06/oracle/oradata/aux1/redo03.log' size 100m;
```

除此之外，还可以使用目标数据库中已有的日志文件定义，并且可以使用 init.ora 文件的 LOG_FILE_NAME_CONVERT 参数简单地将该定义移动到新的位置中。LOG_FILE_NAME_CONVERT 参数与 DB_FILE_NAME_CONVERT 参数的作用相似，因此可以使用成对的字符串来转换日志文件，也可以使用字符串转换操作来修改单个目录的名称，如下所示：

```
log_file_name_convert=('/u02/oracle/oradata/redo01a.dbf',
    '/u03/auxiliary/redo01a.dbf',…)
```

19.2　复制的位置

到目前为止，我们已介绍了复制操作中的较难理解的部分，接下来我们必须考虑辅助实例与目标实例的相对位置。要知道将目标实例复制到相同的服务器与复制到远程服务器是不同的，在执行复制操作前必须理解这些差异。

19.2.1　复制到相同的服务器

我们必须小心处理复制到相同服务器的情况，以便不会影响已经存在的目标数据库。

如果只是简单地生成目标数据库 init.ora 文件的副本，则可以运行下面的代码。

```
duplicate target database to aux1;
```

您会遇到一系列问题和错误。出现这些错误的原因可能是已经运行了一个具有相同名称的实例，同时两个数据库使用的是相同的文件位置。

1. 内存问题

第一个内存问题是数据库名称。Oracle 基于 init.ora 文件中的 DB_NAME 参数值引用服务器上的内存段。因此，Oracle 不允许在同一个系统中运行具有相同 DB_NAME 参数值的两个实例。如果尝试加载具有相同名称的另一个实例，就会得到下面的错误消息：

```
ORA-01102: cannot mount database in EXCLUSIVE mode
```

因此，在同一个系统中执行复制操作时，必须将辅助实例 init.ora 文件中的 DB_NAME 参数值修改为与目标实例不同的数据库名，如下所示：

```
db_name='aux1'
instance_name='aux1'
```

2. 文件位置问题

解决了内存问题后，仍然会存在两个数据库都尝试写入相同文件位置的问题。实际上，有 3 种类型的文件会争用相同名称。如果不考虑文件位置问题，复制操作会在尝试重新构建控制文件的时候失败，如下所示：

```
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of Duplicate Db command at 07/02/2009 13:52:14
RMAN-06136: ORACLE error from auxiliary database:
     ORA-01503: CREATE CONTROLFILE failed
ORA-00200: controlfile could not be created
ORA-00202: controlfile:
'/space/oracle_user/OraHome1/oradata/sun92/control01.ctl'
ORA-27086: skgfglk: unable to lock file - already in use
SVR4 Error: 11: Resource temporarily unavailable
```

其实这对用户来说是好消息，因为这样不会重写产品控制文件。然而必须修改辅助数据库 init.ora 文件中的 CONTROL_FILES 参数以指向磁盘上的新位置，RMAN 通过这种方法确定控制文件的还原位置。

修改控制文件的位置后，还必须修改数据文件的位置。前面已经讨论过，我们可以使用 configure 命令、DB_FILE_NAME_CONVERT 参数或 Oracle8i 样式的 run 代码块这三种方式来修改数据文件的位置。如果向同一个服务器上复制时没有修改数据文件的位置，用户会得到一条与前面控制文件错误相似的错误消息，该消息提示数据文件当前正在使用中而不能被重写。

最后，还必须修改重做日志文件的位置。在前面讨论复制操作的不同步骤时已经介绍过这个内容。我们可以在 duplicate 命令中使用 logfile 关键字来构建完全不同的重做文件，构建的这些重做文件具有不同的大小、组数和成员数，这个选项实际上重写了 create controlfile 命令复制阶段的 logfile 参数。此外，还可以简单地使用辅助 init.ora 文件中的 LOG_FILE_NAME_CONVERT 参数来修改重做日志文件的位置。

19.2.2　复制到相同服务器的不同 ORACLE_HOME 目录

将产品数据库复制到相同服务器上的不同 Oracle 软件安装目录非常常见。当辅助实例具有不同的 ORACLE_HOME 目录时，所遵守的规则会稍微有所不同。上一小节概述了复制到相同服务器需要遵守的所有规则，除此之外，这里还必须考虑备份片的位置。如果复制磁盘备份，只需要提前确认具有相应的 OS 权限。然而，如果复制磁带备份，就必须确认 MML 文件以与目标实例的 ORACLE_HOME 相同的方式链接辅助实例的 ORACLE_HOME，否则辅助实例就不能访问磁带备份，并且由于介质管理器不可访问，复制操作将会失败。

19.2.3　复制到远程服务器

复制到不同的服务器的辅助实例上与复制到相同的服务器上情况的复杂程度相似，不过实现的方法有所不同。

1. 内存问题

与复制到相同服务器的情况不同，我们不必考虑 init.ora 文件中的 DB_NAME 参数问题，因为在不同的服务器上，Oracle 不会限制内存所使用的 LOCK_NAME。

当然，我们最好在复制进程期间注意 DB_NAME 参数，并且在复制操作开始前交叉检查相同服务器上运行的其他实例，这样就不会产生错误。另外，从管理的角度考虑，为您的系统环境中的每个数据库指定一个不同的数据库名是非常明智的举措。

2. 文件位置问题

由于是在一个新的服务器上，所以也不必立即为辅助实例修改文件规范信息。在新的服务器上不会运行具有相同文件的另一个数据库，因此可以保留与目标实例相同的所有文件规范，这样能够避免在配置中出错。同样地，在位于不同的系统中时，我们可以简化许多进程。如果没有修改文件的位置，就必须在 duplicate 命令中指定 nofilenamecheck 参数，该参数通知复制操作不必在执行还原操作前确认文件名是不同的。如果没有指定 nofilenamecheck 参数，则 RMAN 会给出一条错误消息。

对于这种简化的设置来说，需要注意的一个问题是有可能辅助主机不具有与目标主机相同的文件结构和加载点设置。如果加载点或驱动器配置不同，我们仍然需要为辅助实例修改文件规范，这样 RMAN 才能在实际存在的位置执行还原操作。

3. 备份位置为磁盘

在不同服务器上执行还原操作的复杂性在于要提供访问不同服务器上备份的辅助信道进程的访问。对于磁盘备份和磁带备份来说，我们都必须考虑这个问题。

如果要复制磁盘备份，我们的选择会受到限制。前面介绍过，RMAN 将对 DBMS_BACKUP_RESTORE 包的调用传递给辅助实例上的信道进程，但是该操作没有考虑文件系统的差异。RMAN 必须在目标数据库控制文件中记录的实际位置和格式中查找备份片。例如，可以使用下面的命令在目标系统上执行数据库的完整备份操作：

```
backup database format= '/u04/backup/prod/%U.fuu.PROD';
```

执行这条命令会在/u04/backup/prod 目录中创建一个名为 01DSGVLT_1_1 的文件(即备份片)，并且在目标数据库控制文件中记录这个备份。在随后的复制操作期间，RMAN 向辅助实例传递 file restore 命令，并且通知辅助实例从/u04/backup/prod/01DSGVLT_1_1 中还原文件。这就意味着在辅助实例上必须存在一个名为/u04 的加载点和一个名为 backup/prod 的目录，并且在这个目录中驻留一个名为 01DSGVLT_1_1 的文件。如果不存在这些元素，复制操作就会失败，同时给出下面的错误消息：

```
RMAN-03002: failure of Duplicate Db command at 07/02/2002 14:49:55
RMAN-03015: error occurred in stored script Memory Script
ORA-19505: failed to identify file "/u04/backup/prod/01dsgvlt_1_1"
ORA-27037: unable to obtain file status
SVR4 Error: 2: No such file or directory
Additional information: 3
```

可以使用 3 种方法来复制磁盘备份。第一种方法是将备份从目标主机复制到辅助主机的相同位置，该方法非常直观。显然，这种方法要通过网络大量地传输文件。

第二种方法是从辅助主机中 NFS 加载(NFS-mount)目标主机上的备份位置。只有当加载目标的位置与 RMAN 将使用的加载点名称相同的情况下(例如在前面的示例中，必须 NFS 加载/u04/backup/prod 为/u04/backup/prod)，才能执行复制操作。例如，我们需要从辅助实例中执行下面的加载命令：

```
mount cervantes:/u04/backup/prod /u04/backup/prod
```

这样，就能够从辅助节点执行下面的命令：

```
cd /u04/backup/prod
ls -l
touch testfile
ls -l
```

在尝试改变目录或者尝试接触一个文件时，如果由于某种原因得到一条错误消息，就必须在执行复制操作前解决 NFS 和权限问题。图 19-2 展示了通过加载的文件系统将磁盘备份复制到不同的服务器上的方法。

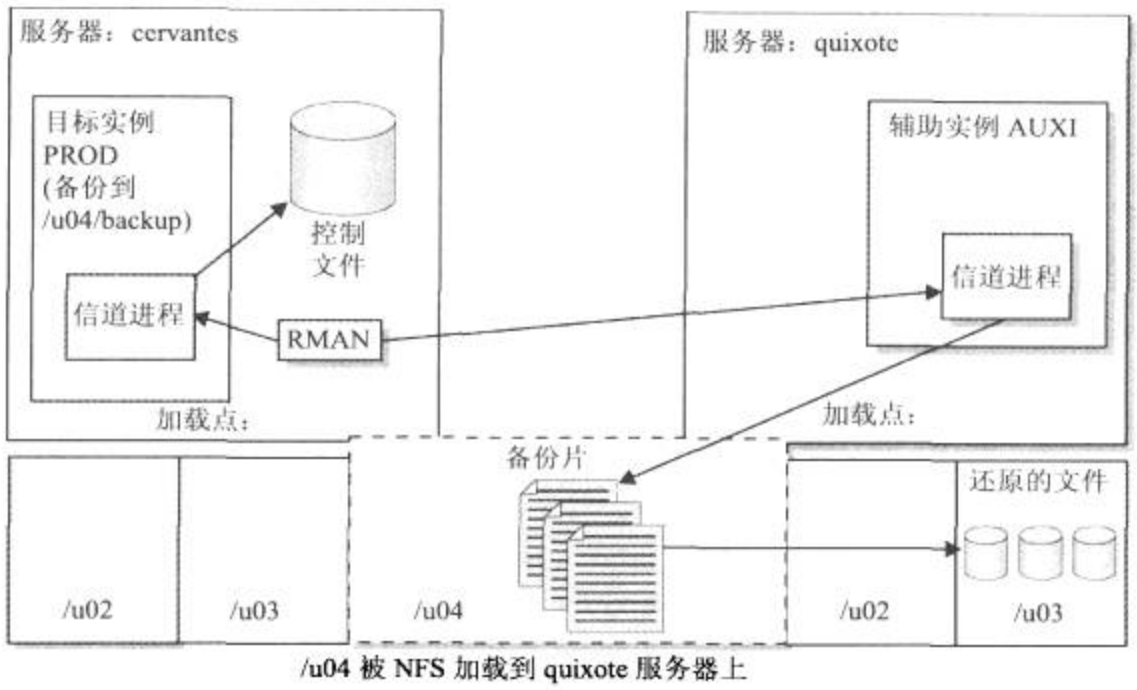


图 19-2　将磁盘备份复制到不同的服务器上

如果使用 Windows 平台，我们加载的就是网络驱动器而不是 NFS，此时应用的规则是相同的：辅助主机与目标主机上的驱动器规范必须相同。因此，如果备份被写入 F:\backup，就必须能够将 F:用作网络驱动器，否则复制操作会失败。此外，必须设置辅助服务(oracleserviceaux1)和侦听程序服务(oracleOraHome92tnslistener)，并以具有读/写权限的域管理员身份登录辅助主机和目标主机。否则，就不能通过网络驱动器访问备份。

您可能已经注意到，成功地执行网络文件系统操作比较困难。如果目标服务器和辅助服务器具有相同的文件系统，就不能直接从辅助服务器中 NFS 加载目标服务器的磁盘备份位置。因此，我们只能选择将备份片从一个节点复制到另一个节点。

出现这些问题的真正原因是 RMAN 会在执行磁盘备份操作时硬编码备份位置，而且这个硬编码的位置不能被修改。然而，在 Oracle Database 10g 中，为用户提供了两个修改备份位置的选项：backup backupset 命令和 catalog backupset 命令。

使用 backup backupset 命令可以备份磁盘上的前一个备份集，并且将其移动到不同的磁盘位置。这就提供了相当大的灵活性。我们现在可以将备份片从/u04/backup/prod 移动到/u06/backup/prod，随后可以从辅助系统中 NFS 加载/u06/backup/prod。还可以从目标主机 NFS 加载辅助主机上的一个驱动器，然后再使用 backup backupset 命令将备份移动到辅助主机上。第 11 章已详细介绍过 backup backupset 命令。

catalog backupset(以及 catalog datafilecopy)命令提供了重新定位新服务器上的备份集的另一种简单方法。为了使 RMAN 了解到备份集所在的位置，只需要告诉 RMAN 对指定的文件(或指定的目录)编写目录，它就会在该位置查找任何有效的备份并且为其生成元数据。查看第 16 章以了解 catalog 命令的更多信息。

4. 备份位置为磁带

总的来说，将磁带备份复制到远程服务器比将磁盘备份复制到远程服务器容易得多，这是由于磁带备份本身只具有文件句柄而不具有文件位置。RMAN 只关心磁带备份的文

件句柄，而介质管理器完全控制文件句柄与具体磁带位置的联系。因此，复制磁带备份时使用的所有配置都来自于介质管理层的配置。

首先，必须在辅助站点上配置 MML 文件，这个配置与目标站点上的配置相同。由于辅助信道执行还原操作，它必须能够如第 4 章所介绍的那样初始化 MML，因此要确保在辅助站点上链接 MML。

接下来，必须确认正确地配置了介质管理服务器，也就是说必须在目标节点所注册的同一个介质管理服务器中将辅助节点注册为一个客户端，并且辅助站点必须具有为了还原而能够访问磁带的必要权限。最重要的是，必须支持辅助节点还原不同服务器上生成的备份。由于能够安全地将一个客户端上的文件还原到另一个客户端是一个潜在的安全漏洞，大多数介质管理软件默认禁用这个功能。根据不同的软件供应商，我们可以按照 4 个介绍介质管理的章节(第 5 章～第 8 章)所概述的步骤启用在不同客户端上还原文件的功能。

配置完介质管理服务器后，最后一个配置步骤是设置辅助信道。前面介绍过，RMAN 在辅助实例上分配一个或多个信道来执行复制操作中的还原与恢复步骤。从 RMAN 连接到目标数据库时，我们可以使用 configure 命令配置这些信道。用于辅助信道的参数 parms 必须包含通常的介质管理环境控制变量，尤其需要指定生成备份的客户端。例如，目标节点名为 cervantes，辅助节点名为 quixote。由于是从客户端 cervantes 备份，这个客户端名与 RMAN 备份一起被编码在介质管理服务器上，所以为了能够从客户端 quixote 访问这些备份，我们必须在 RMAN 中将生成备份的客户端名指定为 cervantes。辅助信道的配置命令如下所示(用于 NetBackup media management system)：

```
RMAN> configure auxiliary channel 1 device type sbt parms
2> = "env=(nb_ora_serv=mgtserv, nb_ora_client=cervantes)";
new RMAN configuration parameters:
CONFIGURE AUXILIARY CHANNEL 1 DEVICE TYPE 'SBT_TAPE' PARMS
"env=(nb_ora_serv=mgtserv, nb_ora_client=cervantes)";
new RMAN configuration parameters are successfully stored
```

执行上面的配置命令，在辅助信道生成对介质管理层的 sbt( )调用时，介质管理服务器会访问客户端 cervantes 生成的备份，而不是访问客户端 quixote 的备份。

19.2.4 复制和网络

前面介绍了几乎所有与复制操作有关的复杂问题，不过在从 RMAN 运行 duplicate 命令之前还必须讨论网络的问题。提到网络，通常是指配置 Oracle Net 文件：tnsnames.ora 和 listener.ora 文件。不过，也可借机审视整个网络并确保目标节点、辅助节点和介质管理服务器之间能够相互访问以及网络具有足够的带宽。

从 Oracle 的观点出发，我们必须配置 Oracle Net 文件。第 2 章曾经讨论过，RMAN 必须生成到目标数据库的 SYSDBA 连接。如果使用远程连接，就必须为目标节点配置口令文件。此外，需要使用一个 tns 别名，它使用专用的服务器进程，而不是共享的服务器进程。对于复制操作来说，同样需要遵守这些规则，但必须只使用专用服务器来生成

到辅助实例的 SYSDBA 连接。

这就意味着不管在何种情况下，都必须为目标机器或辅助机器创建口令文件。您可能到目前为止一直都是本地连接目标数据库，但是不能同时生成到目标实例和辅助实例的本地连接，因此，如果以前没有创建口令文件，现在就应当构建这个文件。

RMAN 实习：构建一个口令文件

实习注意事项

在 UNIX 平台上，口令文件名必须为 orapw<sid>，其中 sid 是口令访问的实例的 ORACLE_SID 值。在下面的实习中，ORACLE_SID=prod。在 Windows 平台上，口令文件名的格式必须为 pwd<sid>.ora。我们必须使用本实习给定的位置创建口令文件，否则这个口令文件不可用。

步骤 1　编辑 init.ora 文件并添加下面的参数:

```
remote_login_passwordfile=exclusive
```

如果使用服务器参数文件，则需要执行如下命令:

```
alter system set remote_login_passwordfile=exclusive scope=spfile;
```

这两个操作都需要重新启动数据库。

步骤 2　确定要创建的口令文件，然后定位到 ORACLE_HOME/dbs 目录(或者是 Windows 平台上的 ORACLE_HOME/database 目录)，输入下面的命令:

```
orapwd file=orapwprod password=<OraclE4ever>
```

步骤 3　检查成功地创建了口令文件，然后通过生成一个远程的 SYSDBA 连接测试该文件。

为辅助实例创建口令文件后，还必须配置侦听程序，把接入连接路由给辅助实例。您可能会注意到：在 Oracle Database 10g 或 11g 中，如果只连接打开数据库，就不需要 listener.ora 文件，这是因为数据库的 PMON 进程使用系统上运行的侦听守护程序自动注册该数据库。因此，即使在 11gR2 的默认安装后没有执行任何配置操作，我们经常能看见一个正在运行的、侦听数据库的侦听程序。

由于必须能够远程连接启动但未打开(即未加载模式)的辅助实例，所以这个功能在复制环境中没有什么用处。由于辅助实例未打开，此时不存在使用侦听程序注册辅助实例的 PMON 进程，所以侦听程序完全不知道辅助实例的存在。为了解决这个问题，必须在旧样式的 listener.ora 文件中手动设置一个辅助实例的条目。建议您使用 Oracle Net Manager 实用程序来构建这个条目。如图 19-3 所示。

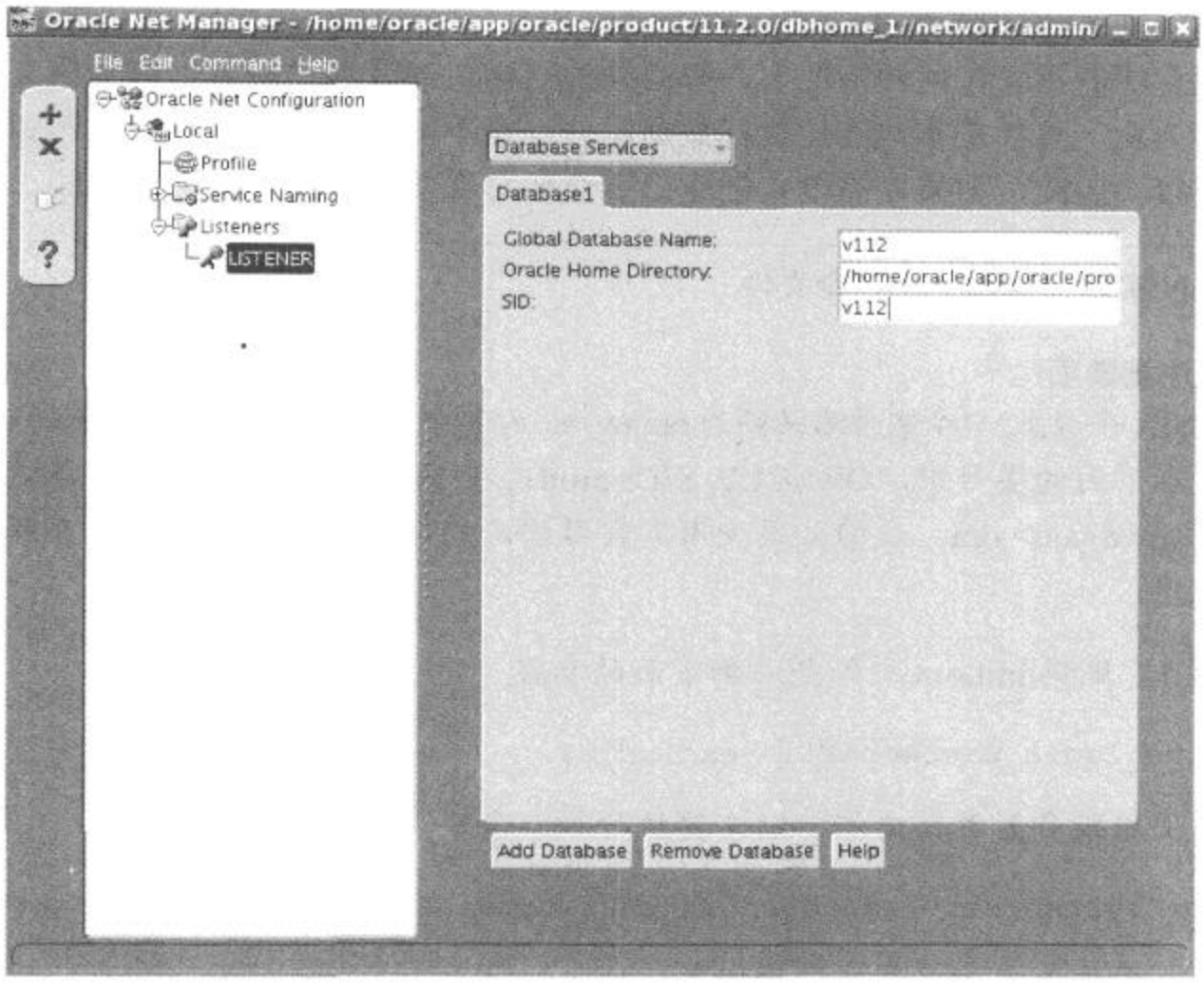


图 19-3 使用 Net Manager 为辅助实例配置一个条目

在辅助实例位置上配置 listener.ora 文件后，还必须在将运行 RMAN 的站点上构建一个 tnsnames.ora 条目。tnsnames.ora 条目与其他条目几乎完全相同，不过在构建 tnsnames.ora 条目时，必须指定辅助实例的 SID_NAME，而不是 SERVICE_NAME。在 Oracle Net Manager 中单击 Service Naming 激活 Net Service Name Wizard，然后进入菜单并选择 Edit | Create 命令，指定了 Net Service Name(第 1 步)、协议(第 2 步)、主机名和端口号(第 3 步)后，指定 Service Name 然后结束，参见图 19-4。

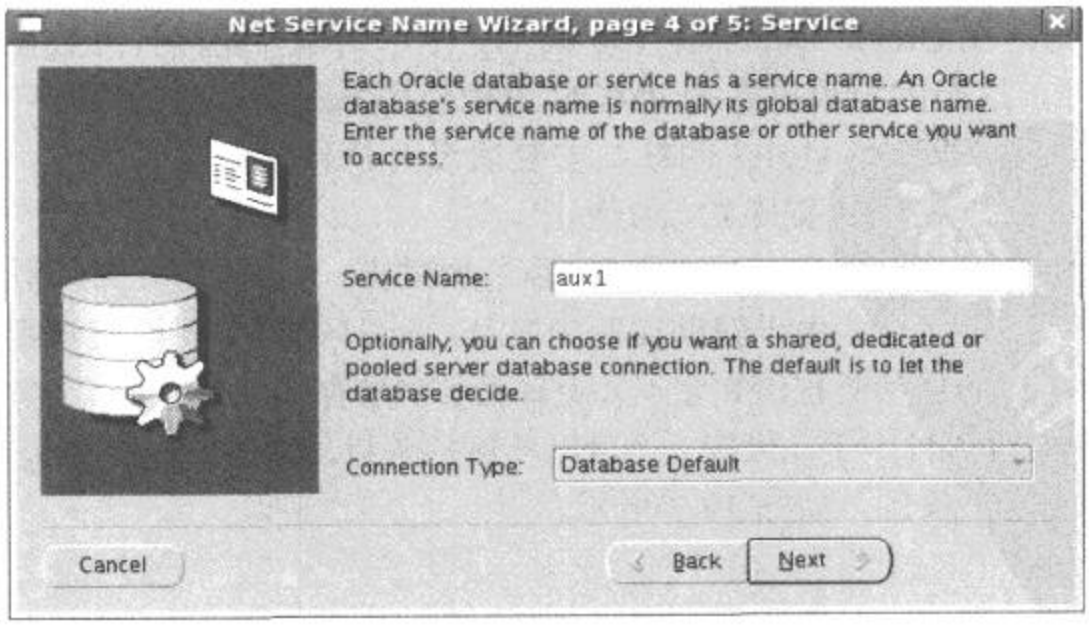


图 19-4 指定辅助实例的 SID

19.3　在相同服务器上复制

做完所有准备工作后，就可以开始进行复制。首先，将详细介绍一个在目标实例驻留的相同服务器上复制磁盘备份的实习，然后再简要介绍在相同服务器上复制磁带备份的操作。

在复制时设置 until 子句

执行复制操作时，有时需要指定 until 子句。如果曾使用了备份的控制文件还原数据库并且现在要复制这个数据库，就会要求设置一个 until 子句。我们建议您先确定复制操作可用的最新归档日志，然后在复制操作期间使用这个日志作为结束点，如下所示：

```
run { set until sequence n thread 1;
duplicate target database… }
```

或者，可以将 SCN 设置为一个不可能出现的大数值，如下所示：

```
set until scn  281474976710655;
```

RMAN 实习：在相同的服务器上复制磁盘备份

实习注意事项

确认配置操作系统能够处理另一个 Oracle 实例，并且确保有足够的内存和磁盘空间。在下面的示例中，目标数据库为 v112，这个数据库的所有数据文件、控制文件和重做日志文件都被存储在/u01/app/product/oradata/v112 中。所有备份都存储到位于/u01/app/oracle/product/flash_recovery_area 的本地闪回恢复区(FRA)中。我们还要将辅助实例使用的 ORACLE_SID 设置为 aux1。

步骤 1　构建辅助数据库目录结构：

```
$ pwd
/u01/app/oracle/product/oradata
$ mkdir aux1
$ mkdir aux1/arch
$ cd ../admin
$ mkdir aux1
$ cd aux1
$ mkdir pfile bdump udump cdump
$ ls
  bdump cdump pfile udump
```

步骤 2　将目标数据库的 init.ora 文件复制到辅助数据库的相应位置上。如果目标数据库使用了服务器参数文件(SPFILE)，为了获取要移动的参数，就必须从这个服务器参数文件中创建一个参数文件(PFILE)。

如果在目标数据库上使用了服务器参数文件，就输入下面的命令。

```
SQL> connect / as sysdba
Connected.
SQL> create pfile='/u01/app/oracle/product/admin/aux1/pfile/init.ora'
  from spfile;
```

如果在目标数据库上使用了 init.ora 文件，就输入下面的命令：

```
cp u01/app/oracle/product/admin/v112/pfile/init.ora
   u01/app/oracle/product/admin/aux1/pfile/init.ora
```

步骤 3 根据需要修改 aux1 中的 init.ora 文件：

```
audit_file_dest='/u01/app/oracle/product/admin/aux1/adump'
background_dump_dest='/u01/app/oracle/product/admin/aux1/bdump'
compatible='11.2.0.1.0'
control_files='/u01/app/oracle/product/oradata/aux1/control01.ctl',
  '/u01/app/oracle/product/oradata/aux1/control02.ctl',
  '/u01/app/oracle/product/oradata/aux1/control03.ctl'
core_dump_dest='/u01/app/oracle/product/admin/aux1/cdump'
db_block_size=8192
db_file_multiblock_read_count=16
db_name='aux1'
db_recovery_file_dest='/u01/app/oracle/product/flash_recovery_area'
db_recovery_file_dest_size=4294967296
dispatchers='(PROTOCOL=TCP) (SERVICE=aux1XDB)'
job_queue_processes=10
open_cursors=300
pga_aggregate_target=93323264
processes=150
remote_login_passwordfile='EXCLUSIVE'
sga_target=279969792
undo_management='AUTO'
undo_tablespace='UNDOTBS1'
user_dump_dest='/u01/app/oracle/product/admin/aux1/udump'
db_file_name_convert=('v102','aux1')
instance_name='aux1'
```

步骤 4 为辅助实例创建 aux1 口令文件。参阅本章 19.2.4 小节的“构建一个口令文件”实习。

步骤 5 以未加载模式启动 aux1 实例：

```
ORACLE_SID=aux1
export ORACLE_SID
sqlplus /nolog
sql>connect / as sysdba
SQL> startup nomount
pfile=/u01/app/oracle/product/admin/aux1/pfile/init.ora
```

步骤 6 配置连接 aux1 实例的网络文件。修改 listener.ora 文件后，确认重新运行侦听程序，否则修改不会起作用。

```
$ lsnrctl reload
```

tnsnames.ora 文件应当具有下面的条目：

```
AUX1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = horatio)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = aux1)
      (SERVER = DEDICATED)
    )
)
```

listener.ora 文件应当具有下面的条目：

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = aux1)
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME = aux1)
  )
)
```

步骤 7　从 RMAN 连接到目标实例和辅助实例，并且运行 duplicate 命令：

```
$ ORACLE_SID=aux1;export ORACLE_SID
$ rman target /
RMAN> connect auxiliary sys/ora10g@aux1
RMAN> duplicate target database to aux1
  pfile=/u01/app/oracle/product/admin/aux1/pfile/init.ora
  logfile
  '/u01/app/oracle/product/oradata/aux1/redo01.dbf' size 100m,
  '/u01/app/oracle/product/oradata/aux1/redo02.dbf' size 100m,
  '/u01/app/oracle/product/oradata/aux1/redo03.dbf' size 100m;
```

使用磁带备份

如果要在相同的服务器上复制磁带备份，只需对前面介绍的 RMAN 实习稍做修改。实际上，不需要修改任何代码，我们只需要在运行 duplicate 命令前插入一个额外的步骤即可。该步骤配置辅助信道，使之与执行备份的信道类似。换句话说，就是执行下面的 show 命令：

```
RMAN> show channel;
RMAN configuration parameters are:
CONFIGURE CHANNEL 1 DEVICE TYPE 'SBT_TAPE' PARMS
"env=(nb_ora_serv=mgtserv)";
CONFIGURE CHANNEL 2 DEVICE TYPE 'SBT_TAPE' PARMS
"env=(nb_ora_serv=mgtserv)";
```

随后，执行下面的命令创建相匹配的辅助信道：

```
CONFIGURE AUXILIARY CHANNEL 1 DEVICE TYPE 'SBT_TAPE' PARMS
"env=(nb_ora_serv=mgtserv)";
CONFIGURE AUXILIARY CHANNEL 2 DEVICE TYPE 'SBT_TAPE' PARMS
"env=(nb_ora_serv=mgtserv)";
```

19.4 在远程服务器上复制

在远程服务器上执行复制操作具有许多与在相同服务器上执行复制操作相同的配置步骤，特别是在远程复制磁盘备份的情况下，配置步骤是完全相同的，虽然您可以放弃所有文件重命名步骤。此外，既可以将备份复制(copy)到远程服务器，也可以在远程站点上 NFS 加载这些备份。由于本书不讨论 NFS 的有关内容，所以我们假定在下面的 RMAN 实习中，目标服务器和辅助服务器具有相同的文件系统，这样就可以将备份复制到辅助系统中。

RMAN 实习：在远程服务器上复制磁盘备份

实习注意事项

本实习假定存在两个服务器：目标服务器 dex 和辅助服务器 horatio，还假定目标节点和辅助节点具有相同的文件系统，我们将备份从 dex 服务器复制到 horatio 服务器。在这里重点要注意的是，与在相同服务器上执行的复制操作不同，我们在整个过程中都保持 v112 数据库的 SID 不变(并没有将这个数据库的 SID 修改为 aux1)。

步骤 1 在 horatio 服务器(辅助服务器)上，构建辅助数据库目录结构：

```
$> pwd
/u01/app/oracle/oradata
$> mkdir v112
$> cd ../admin
$> pwd
/u01/app/oracle/admin
$> mkdir v112
$> cd v112
$> mkdir pfile bdump udump cdump adump
$> ls
adump  bdump  cdump  pfile  udump
```

步骤 2 在 dex 服务器上(目标服务器)，生成目标数据库 init.ora 文件的副本，以便将 init.ora 文件移动到辅助服务器上。如果目标数据库使用了服务器参数文件(SPFILE)，为了获取要移动的参数，就必须从这个服务器参数文件中创建一个参数文件(PFILE)。

如果在目标服务器上使用了服务器参数文件，就输入下面的命令：

```
SQL> connect / as sysdba
Connected.
```

```
SQL> create pfile='/home/oracle/scratchpad/init.ora'
  from spfile;
```

如果在目标数据库上使用了 init.ora 文件，就要输入下面的命令:

```
cp /u01/app/oracle/admin/v112/pfile/init.ora*
   /home/oracle/scratchpad/init.ora
```

步骤 3　将目标数据库的 init.ora 文件移动到辅助站点:

```
cd /home/oracle/scratchpad/
ftp horatio.hadba.com
username: oracle
password:
cd /u01/app/oracle/admin/v112/pfile
put init.ora
exit
```

另外，也需要一个位于目标数据库 dex 上的 init.ora 文件的本地副本，RMAN 在 duplicate 命令中引用该副本。在第 19 步中运行 duplicate 命令时，我们将引用保存在/home/oracle/scratchpad/init.ora 中的该副本。

步骤 4　在 quixote 服务器上，以未加载模式启动辅助实例:

```
ORACLE_SID=v112; export ORACLE_SID
sqlplus /nolog
SQL>connect / as sysdba
SQL>startup nomount
pfile=/u01/app/oracle/admin/v112/pfile/init.ora
```

步骤 5　在辅助站点(proto)上配置 listener.ora 文件:

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = v112)
      (ORACLE_HOME = /uol/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME = v112)
  )
)
```

步骤 6　配置位于目标站点(dex)上的 tnsnames.ora 文件。

```
V112_HORATIO =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = horatio)(PORT = 1522))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = v112)
    )
```

步骤 7 在远程服务器(proto)上创建口令文件。遵循 19.2.4 小节的实习“构建一个口令文件”中的指令。

步骤 8 将 FRA 文件从 dex 移动到 horatio。

步骤 9 从目标系统(dex 服务器)运行 duplicate 命令:

```
ORACLE_SID=v112; export ORACLE_SID
rman target /
RMAN> connect auxiliary sys/password@v112_horatio
duplicate target database to v112
pfile=/home/oracle/scratchpad/init.ora
logfile
'/app/oracle/product/11.2.0/oradata/v102/redo01.dbf' size 100m,
'/app/oracle/product/11.2.0/oradata/v102/redo02.dbf' size 100m,
'/app/oracle/product/11.2.0/oradata/v102/redo03.dbf' size 100m;
```

在远程服务器上复制磁带备份

如果要在远程服务器上复制磁带备份，就可以应用前面实习中的所有步骤，同样地，还必须配置辅助信道以反映介质管理器的需求。除了指定介质管理服务器和用于常规信道的类或池外，还必须指定目标客户端名，如下所示:

```
RMAN> configure auxiliary channel 1 device type sbt parms
2> = "env=(nb_ora_serv=mgtserv, nb_ora_client=dex)";
```

19.5 11g 的无目标复制

从 11gR2 开始，RMAN 去除了一些先前版本中使复制过程过分复杂的关联。最主要的变化是，从 RMAN 备份复制数据库不需要创建到目标数据库的连接。提供到活动的目标数据库的访问可以使复制的某些方面简化，有时却很难达到。事实上，我们陷入了“伪造目标数据库(fake target database)”操作中，从不可访问的目标数据库中请求备份控制文件，加载控制文件，指示该实例上的 RMAN 开始复制。Oracle 也许是体验到这种痛苦。

无目标复制，下一个最好的办法是连接到恢复目录，可以方便地获取需要的目标数据库元数据。为实现该目的主要的工作是在执行复制操作时关注数据库 ID(DBID)。当连接到目标数据库时，RMAN 不难理解“prod”的含义。如果在目录中有多个“prod”数据库，就可能产生模棱两可的问题。因此，应当深入到恢复目录中获得 DBID:

```
SQL> select dbid, name, resetlogs_time from re_database;
    DBID  NAME       RESETLOGS
    ----------  ---------  ---------
    2203908660  V112    20-JUN-09
```

一旦知道了 DBID，就可以开始与前述的实习方法类似的无目标复制。

```
RMAN> connect auxiliary sys/password@v112_horatio
```

```
Connect catalog rman2/rman2@rcv_cat
duplicate database v112 to aux1
dbid 2203908660
until time "to_date('2009-07-08', 'YYYY-MM-DD')"
pfile=/home/oracle/scratchpad/init.ora
logfile
'/u01/app/oracle/11.2.0/oradata/v102/redo01.dbf' size 100m,
'/u01/app/oracle/11.2.0/oradata/v102/redo02.dbf' size 100m,
'/u01/app/oracle/11.2.0/oradata/v102/redo03.dbf' size 100m;
```

使用 DBNEWID 实用程序的不完全复制

复制时最令人不满意的一点是不存在“可重启的复制”，也就是说，如果在执行还原所有文件的步骤(这是复制进程中最长的步骤)时出错，比如在恢复操作期间出错，就必须再次启动复制进程并重新还原文件。我们无法修复恢复进程(例如使丢失的归档日志可用)，只能重新开始整个进程。

将 RESTORE OPTIMIZATION 选项设置为 ON，RMAN 就不会在还原位置中再次还原具有相同数据文件头 SCN 信息的文件。这也适用于复制操作：如果复制操作还原一个文件，然后重新启动该复制操作，RMAN 将不会再次还原该文件。然后，如果将一个归档日志应用于该文件，则会再次还原该归档日志。

在 Oracle 9*i* Release2 中，使用 DBNEWID 实用程序可以部分修复恢复进程，这个实用程序以安全和可控的方式为复制的数据库指定新的 DBID。我们可以手动恢复复制的数据库，准备所有元素，然后再运行 DBNEWID 实用程序完成由复制操作启动的进程。这就允许我们至少可以手动完成复制操作。

DBNEWID 实用程序的用法非常简单。首先，必须确认在使用这个实用程序前生成了有效的备份，尽管要执行验证备份的进程，但是在修改 DBID 期间仍然可能会遇到不可恢复的错误。确认备份有效后，还必须在一致的状态中关闭数据库，然后将数据库切换回加载状态：

```
shutdown immediate;
startup mount
```

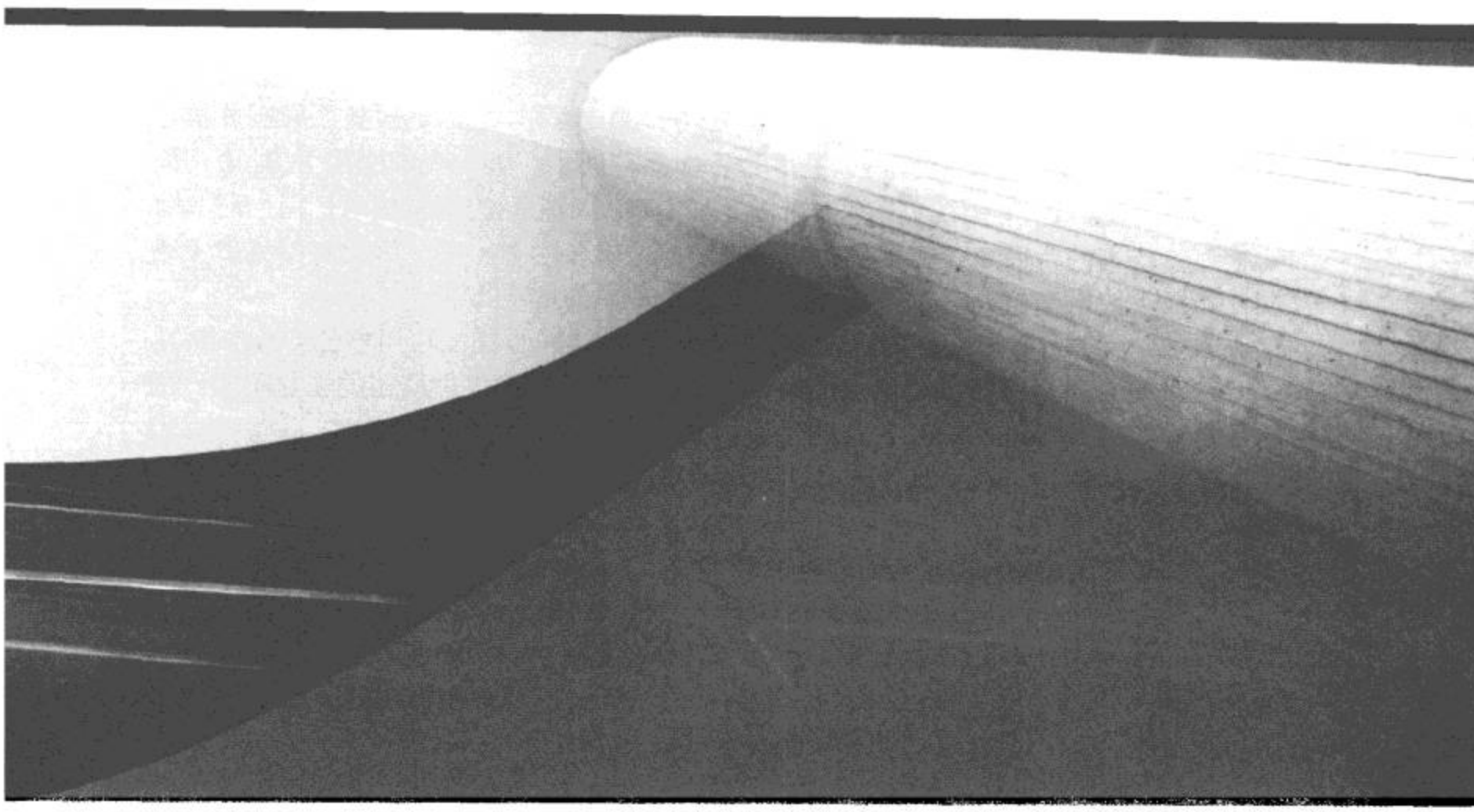
接下来如下所示，从命令行运行 DBNEWID 实用程序：

```
nid target =/
DBNEWID: Release 10.2.0.1.0 - Production on Sat Sep 3 14:06:43 2005
Copyright (c) 1982, 2005, Oracle. All rights reserved.
Connected to database AUX1 (DBID=366194736)
Connected to server version 10.2.0
Control Files in database:
  /u01/app/oracle/product/oradata/aux1/control01.ctl
  /u01/app/oracle/product/oradata/aux1/control02.ctl
  /u01/app/oracle/product/oradata/aux1/control03.ctl
Change database ID of database AUX1? (Y/[N]) => Y
```

```
Proceeding with operation
Changing database ID from 366194736 to 366203699
   Control File /u01/app/oracle/product/oradata/aux1/control01.ctl
-modified
   Control File /u01/app/oracle/product/oradata/aux1/control02.ctl
- modified
   Control File /u01/app/oracle/product/oradata/aux1/control03.ctl
- modified
   Datafile /u01/app/oracle/product/oradata/aux1/system01.dbf
- dbid changed
   Datafile /u01/app/oracle/product/oradata/aux1/undotbs01.dbf
- dbid changed
   Datafile /u01/app/oracle/product/oradata/aux1/sysaux01.dbf
- dbid changed
   Datafile /u01/app/oracle/product/oradata/aux1/users01.dbf
- dbid changed
   Datafile /u01/app/oracle/product/oradata/aux1/temp01.dbf
- dbid changed
   Control File /u01/app/oracle/product/oradata/aux1/control01.ctl
- dbid changed
   Control File /u01/app/oracle/product/oradata/aux1/control02.ctl
- dbid changed
   Control File /u01/app/oracle/product/oradata/aux1/control03.ctl
 - dbid changed
   Instance shut down
Database ID for database AUX1 changed to 366203699.
All previous backups and archived redo logs for this database are unusable.
Database is not aware of previous backups and archived logs in Recovery
Area.
Database has been shutdown, open database with RESETLOGS option.
Successfully changed database ID.
DBNEWID - Completed successfully.

[oracle@dex oracle]$ sqlplus "/ as sysdba"

SQL> alter database open resetlogs;
```

19.6 小结

本章讨论了 RMAN 复制过程的体系结构。通过使用复制操作，可以在本地系统或远程服务器中生成来自于 RMAN 备份的复制数据库，根据执行的是本地复制操作还是远程复制操作，相应的配置步骤是不同的。此外，如果使用磁盘备份或磁带备份，需要遵守特定的规则。我们给出了演示复制操作的两个 RMAN 实习，最后还简要讨论了 Oracle Database 10g 及其后提供的 DBNEWID 实用程序。



第 20 章

RMAN 与 Data Guard

许多 DBA 都非常熟悉被称为 Oracle Standby Database 的功能。Oracle Standby Database 是一个体系结构特性，它针对系统上已经发生的操作添加了一个小功能：即对数据库上发生的每一个变化进行归档。Standby Database 的基本原理非常简单，即使整体的体系结构现在称为 Data Guard，其功能也很简单：即从产品数据库中获得归档的日志，将其移动到具有该数据库的副本的另一台计算机，并且将归档的日志应用于该副本。通过这种方法，Data Guard 就能够通过在远程站点上维持事务上一致的产品数据库副本来提供有效的灾难恢复解决方案。这些副本或备用数据库可以有两种类型：物理的和逻辑的。根据业务需求的不同，可以选择在 Data Guard 配置中应用哪种类型。

通过使用介质恢复来应用在主数据库上生成的重做，物理备用数据库与主数据库保持同步。因为使用了介质恢复，可以确保物理备用数据库是主数据库的块对块对等副本。

由于这种特性，物理备用数据库是解决灾难恢复问题的绝佳选择。在出现故障的情况下，我们可以放心，数据将会完整无缺，并且这些数据与主数据库上的数据保持一致。

通过将从主数据库接收的重做数据转换为逻辑 SQL 语句，然后针对备用数据库执行这些 SQL 语句，逻辑备用数据库可以与主数据库保持同步。因为我们应用 SQL 语句而不是执行介质恢复，逻辑备用数据库可以包含与主数据库相同的逻辑信息，但同时具有不同的物理结构。因为用户在应用更改时可以访问逻辑备用数据库，因此对于维护灾难恢复属性和报告数据库来说，逻辑备用数据库都是理想的解决方案。

集成 RMAN 的 Data Guard 可以生成完整的用于灾难恢复和高可用性的 Oracle 解决方案。RMAN 备份可以用于创建底层的备用数据库，同时还能提供最初的恢复阶段。创建备用数据库并配置 Data Guard 代理程序后，RMAN 可连接到备用数据库，获得能够用于还原主数据库的备份。通过这种方式，我们可以从产品环境中完全删除执行备份所使用的资源。

本书主要讨论与 RMAN 有关的内容，对 Data Guard 的介绍相对较少。如果您要进一步了解 Oracle Standby Database 或 Data Guard，可以参阅 Oracle Press 出版的关于 Data Guard 方面的书籍(由 Larry Carpenter 与公司出版的最新版本非常精彩)。在接下来的章节中，我们假定您已经掌握了 Data Guard 备用数据库的基础知识，并且准备好了使用 RMAN 创建备用数据库。

20.1 RMAN 与备用数据库

首先，RMAN 与备用数据库之间的关系非常简单：RMAN 用于创建备用数据库。如果应用的备份策略是让 RMAN 执行所有备份操作，那么除了选择使用 RMAN 外别无选择，因为必须先将主数据库的数据文件复制到备用数据库，然后备用数据库才能开始发挥作用。

自 Oracle Database 10*g* 版本开始，RMAN 备份可用于创建物理备用数据库和逻辑备用数据库。在 Oracle 9*i* Database 中，在将备份用于逻辑备用之前，主数据库必须是静态的(quiesced)。而在 Oracle Database 11*g* 中，热备份可用作逻辑备用数据库的基础，这意味着现在可以使用 RMAN 备份。

然而，如果阅读过《Oracle Database 10*g* RMAN 备份与恢复》一书，您就会发现在指令方面并没有太多的改动。这是因为 RMAN 并不区分物理备用数据库和逻辑备用数据库的创建。RMAN 不进行这种区分是因为：创建逻辑备用数据库的第一步就是创建物理备用数据库。创建物理备用数据库之后，将其转换为逻辑备用数据库。RMAN 只进行物理备用数据库的创建，而将逻辑备用数据库的创建工作留给用户完成。

20.1.1 使用 RMAN 创建备用数据库的必要条件

使用 RMAN 创建备用数据库前，我们必须执行一些初始的步骤，这些步骤与第 17 章概述的复制进程步骤相同：

(1) 对主数据库和备用数据库中的 init.ora 文件进行必要的修改。

(2) 完成所有必要的 Oracle Net 配置。

(3) 以 NOMOUNT 模式启动备用数据库。

初始步骤相同的原因是 RMAN 使用 duplicate 命令创建备用数据库，因此创建备用数据库应用的所有规则也都应用到复制操作上。也就是说创建备用数据库与执行复制操作时，文件重命名的问题、口令文件的需求、配置 listener.ora 和 tnsnames.ora 文件的方法都是相同的。您可以参阅第 19 章所介绍的复制操作的准备工作，从本小节开始介绍的是除第 19 章介绍的必要条件和约束之外，创建备用数据库还有哪些要求和限制。

除复习第 19 章的内容外，还必须生成特定类别的控制文件备份。由于备用数据库的控制文件与常规的控制文件不同，所以创建备用数据库时不会使用常规的控制文件备份。它对于在介质恢复操作期间应用归档日志有更严格的条件。备用控制文件也不允许使用 alter database open 命令打开备用数据库，而是必须使用 alter database activate standby database 命令激活备用数据库，并且使这个备用数据库成为主数据库。

使用下面的 SQL 命令可以创建一个备用控制文件：

```
alter database create standby controlfile as
    '/u04/backup/stby_cfile01.ctl';
```

如果要以这样的方式创建备用控制文件，就必须在 RMAN 能够使用该文件执行 duplicate...for standby 命令前在 RMAN 中为这个备用控制文件编写目录：

```
RMAN>catalog controlfilecopy '/u04/backup/stby_cfile01.ctl';
```

我们还可以简单地使用 RMAN 创建备用控制文件。如下所示，可以使用 backup 命令在磁带上(如果使用磁带备份)创建控制文件的副本：

```
RMAN>  backup current controlfile for standby;
```

创建了备用控制文件后，就可以使用 RMAN 创建备用数据库。

在讨论创建备用数据库的命令之前，我们需要介绍一些术语。前一章介绍过，RMAN 复制进程必须连接的两个数据库是目标数据库和辅助数据库。目标数据库是备份的来源，辅助数据库是根据目标数据库备份创建的数据库副本。

在备用数据库体系结构的术语中，同样也存在两个数据库：一个是主数据库(primary database)，它是当前的产品数据库；另一个是备用数据库(standby database)，它是未打开的主数据库副本，并且接收来自主服务器的归档的重做日志。使用 RMAN 创建备用数据库时，必须记住：目标数据库是主数据库，辅助数据库是备用数据库。

20.1.2　duplicate...for standby 命令

创建备用数据库时，duplicate 命令的用法稍有不同。前面介绍过，创建备用数据库时适用所有复制规则，不过还必须添加一些说明和约束。

1. 内存问题

首先，备用数据库 init.ora 文件中的 DB_NAME 参数必须与主数据库 init.ora 文件中

的 DB_NAME 参数匹配。如果在不同的服务器上创建备用数据库，这个问题并不重要。不过，如果在主数据库所在的相同服务器上创建备用数据库，这个问题就十分重要。此时，Oracle 会使用 DB_NAME 参数在 OS 层上锁定内存段。因此，如果尝试加载具有相同 DB_NAME 参数值的两个实例，就会得到下面的错误信息：

```
ORA-01102: cannot mount database in exclusive mode>
```

要修正这个错误，我们需要修改备用数据库的 init.ora 文件并添加 LOCK_NAME_SPACE 参数。将 LOCK_NAME_SPACE 参数值设置为不同于 DB_NAME 的值，这样 Oracle 不需要修改 DB_NAME 参数值就能使用 LOCK_NAME_SPACE 参数值锁定内存段，如下所示：

```
Lock_name_space='stby'
```

由于 DB_NAME 参数值必须是相同的，所以 duplicate for standby 命令不使用 duplicate target database to <aux_name>的格式。在复制操作期间，我们必须用这种格式重置新控制文件中的数据库名，但是在备用数据库创建期间并不会创建新的控制文件，所以 duplicate for standby 命令的格式应当如下所示：

```
duplicate target database for standby;
```

2. 日志文件问题

在创建备用数据库时，duplicate 命令中除了没有 to 之外，也不能使用 logfile 子句。这与备用数据库选项有关，例如 standby redo logfile 选项是 Data Guard 解决方案中的一部分。这些选项都不应用于复制进程，但由于它们的存在，RMAN 不接受 logfile 子句。如果要直接配置备用数据库并且为日志文件指定不同的位置，就必须在备用数据库的 init.ora 文件中使用 LOG_FILE_NAME_CONVERT 参数，这个参数与 DB_FILE_NAME_CONVERT 参数的作用相似，它是一个简单的字符串变换，可用于单个目录名的改变，也可以被成对地用于多个修改。

3. 数据文件问题

如果要在与保存主数据库不同的服务器上创建备用数据库，只要备用数据库服务器具有与主数据库所在服务器相同的加载点和目录结构，就不必修改与数据文件位置相关的信息。和常规复制一样，如果在备用数据库上使用与主数据库相同的数据文件位置，就必须在 duplicate 命令中使用 nofilenamecheck 关键字，如下所示：

```
duplicate target database for standby nofilenamecheck;
```

如果要在与主数据库相同的服务器上创建备用数据库，就必须修改备用数据库数据文件的位置。我们可以使用与常规复制相同的任何一种方法：使用 configure auxname 命令；在 run 代码块中使用 set newname 命令；以及在备用数据库的 init.ora 文件中使用 DB_FILE_NAME_CONVERT 参数。

4. 介质恢复问题

物理备用数据库不会因读/写操作而被打开。它会被加载和置入介质恢复状态，或者以只读模式被打开，此时只允许用户执行 select 语句。由于这个特性，创建备用数据库所执行的 RMAN 复制操作与克隆数据库稍有不同。首先，RMAN 不重新创建控制文件，而是还原备用控制文件；其次，在还原数据文件后，RMAN 默认不对这些数据文件执行任何介质恢复操作，并且跳过重置 DBID 和打开数据库重置日志的步骤。因此，完成 duplicate...for standby 命令时，备用数据库只被加载但未打开，并且我们需要确定继续执行介质恢复操作的方法。

如果要使 RMAN 执行初始的介质恢复操作和应用 RMAN 备份的归档日志，则可以在 duplicate 命令中使用 dorecover 关键字，如下所示：

```
duplicate target database for standby dorecover;
```

通常，如果要执行恢复操作，尤其是如果要在远程站点上创建备用数据库，就要为 duplicate 命令提供 until 子句。如果我们位于远程站点上，使用 set until time 或 set until sequence 命令时可以确保恢复操作只尝试使用其他站点上可用的归档日志。当然，由于没有设置 DBID 或打开克隆数据库，所以 duplicate 命令中失败的介质恢复会话并不重要，我们只需在停止的位置手动重新开始即可。

RMAN 实习：使用 RMAN 创建备用数据库

实习注意事项

本实习将在与主数据库所在的相同服务器上创建备用数据库。备用数据库的 ORACLE_SID 为 stby，不过 DB_NAME 与主数据库的 DB_NAME 同为 v112。我们将使用 DB_FILE_NAME_CONVERT 和 LOG_FILE_NAME_CONVERT 参数，并且执行介质恢复操作。

步骤 1　使用 RMAN 创建一个备用控制文件：

```
ORACLE_SID=v112;export ORACLE_SID
rman target /
RMAN> backup current controlfile for standby;
```

创建这个备用控制文件后，必须指定一个时间点，所以要执行一些日志切换操作，然后记录 V$ARCHIVED_LOG 视图中的最后一个日志序列号：

```
SQL> alter system switch logfile;
SQL> alter system switch logfile;
SQL> select sequence# from v$archived_log;
```

步骤 2　构建备用数据库目录结构：

```
$ cd $ORACLE_BASE
$ cd oradata
$ pwd
```

```
/u01/app/oracle/oradata
$ mkdir stby
$ cd ../admin
$ pwd
/u01/app/oracle/admin
$ mkdir stby
$ cd stby
$ mkdir pfile adump bdump cdump udump
```

步骤 3 将目标数据库的 init.ora 文件复制到辅助数据库的相应位置上。如果目标数据库使用了服务器参数文件(SPFILE)，为了获取要移动的参数，就必须从这个服务器参数文件中创建一个参数文件(PFILE)。

如果在目标数据库上使用了服务器参数文件，就输入下面的命令:

```
SQL> create pfile='/u01/app/oracle/admin/stby/pfile/init.ora'
  2 from spfile;
```

如果在目标数据库上使用了 init.ora 文件，就输入下面的命令:

```
cp /u01/app/oracle/admin/v112/pfile/init.ora
/u01/app/oracle/product/admin/stby/pfile/init.ora
```

步骤 4 根据需要修改 stby 中的 init.ora 文件:

```
*.audit_file_dest='/u01/app/oracle/admin/stby/adump'
*.background_dump_dest='/u01/app/oracle/admin/stby/bdump'
*.compatible='11.2.0.1.0'
*.control_files='/u01/app/oracle/oradata/stby/control01.ctl',
     '/u01/app/oracle/oradata/stby/control02.ctl',
     '/u01/app/oracle/oradata/stby/control03.ctl'
*.core_dump_dest='/u01/app/oracle/admin/stby/cdump'
*.db_block_size=8192
*.db_domain='' *.db_file_multiblock_read_count=16
*.db_name='v112'
*.db_recovery_file_dest='/u01/app/oracle/product/flash_recovery_area'
*.db_recovery_file_dest_size=4294967296
*.job_queue_processes=10
*.open_cursors=300
*.pga_aggregate_target=93323264
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=279969792
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'
*.user_dump_dest='/u01/app/oracle/admin/stby/udump'
lock_name_space='stby'
db_file_name_convert=('v112','stby') log_file_name_convert=('v112','stby')
lock_name_space='stby'
```

步骤 5　为 stby 创建口令文件。参阅第 19 章 19.2.4 小节的“构建一个口令文件”实习。

步骤 6　以未加载模式启动 stby 实例：

```
ORACLE_SID=stby; export ORACLE_SID
sqlplus "/ as sysdba"
sql>startup nomount
pfile=/u01/app/oracle/admin/stby/pfile/init.ora
```

步骤 7　配置连接 stby 实例的网络文件。修改 listener.ora 文件后，确认重新运行侦听程序，否则修改不会起作用。

```
Lsnrctl reload
```

tnsnames.ora 文件应当具有下面的条目：

```
STBY =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = horatio)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = stby)
    )
)
```

listener.ora 文件应当具有下面的条目：

```
(SID_DESC =
  (GLOBAL_DBNAME = stby)
  (ORACLE_HOME = /u01/app/oracle/product/11.2.0/dbhome_1)
  (SID_NAME = stby)
)
```

步骤 8　从 RMAN 连接到目标实例和辅助实例，并运行 duplicate 命令：

```
ORACLE_SID=v112; export ORACLE_SID
rman target /
RMAN> connect auxiliary sys/password@stby
RMAN>run {
  set until sequence = 789 thread = 1;
  duplicate target database for standby
  dorecover;}
```

20.2　从备用数据库生成备份

创建了备用数据库后，我们可以根据不同的目的使用这个备用数据库。当然，备用数据库存在的主要原因是能够为产品数据库提供一个灾难恢复解决方案。不过，我们也可以暂停备用数据库的介质恢复操作，以只读模式打开这个备用数据库，并且执行任意

数目的数据采集操作，这些操作会占用所生成系统的大量资源。

从 RMAN 的观点出发，使用备用数据库还有一个优秀的方法。第 16 章介绍过，运行 RMAN 会占用更多的资源，执行备份操作会使用宝贵的内存、CPU 和磁盘 I/O 资源。因此，我们建议您在数据库的闲置时间执行备份操作。但是，有些数据库不存在闲置时间。例如，产品数据库可能 1 天 24 小时都在使用，因为它在频繁地更新；或者，由于数据库很大，需要昼夜不停地执行备份操作。

如果存在物理备用数据库，就可以从备用数据库中生成产品数据库备份。如果主数据库出现故障，就可以在主数据库上还原这些备份。由于备用数据库具有与主数据库相同的 DBID，并且通常来自于相同的对应物，所以 RMAN 数据文件备份在备用数据库和主数据库之间可以互换。备用数据库实际上是主数据库的克隆。

使用备用数据库生成产品数据库备份时，需要记住的一点是，RMAN 会将备用数据库作为目标数据库连接。在前面介绍的复制操作中，我们将备用数据库看作是辅助数据库。但是一旦建立了备用数据库，就能够将备用数据库作为目标数据库连接并执行 backup 操作，在主数据库上可以使用这些备份执行还原操作。

要将备用数据库作为目标数据库连接，就必须建立恢复目录。如果不使用恢复目录，就不能将备份记录从备用控制文件传送到主数据库的控制文件中。如果使用了恢复目录，就可以在备份后与备用控制文件再同步，这样备份记录就能被置入恢复目录。随后，将主数据库作为目标数据库连接，并且连接恢复目录。对于 RMAN 来说，主数据库和备用数据库难以区分，不论连接到哪个数据库，RMAN 都会连接到恢复目录中的相同记录集。因此，连接到主数据库时也会执行再同步操作，并且还会使用连接到备用数据库时生成的备份的记录刷新主数据库的控制文件。

使用 UR=A 参数

如果正在使用实例注册，则 listener.ora 文件中默认为没有任何数据库条目。可以在侦听程序中手动配置数据库条目，或者可以使用新参数 UR=A 更新 tnsnames.ora 文件。该参数允许只进行管理性的连接，如下所示：

```
CONNECT_DATA=((SID=....) (UR=A))
```

在 11g 中，可以通过从使用 pfile 变为使用 spfile，以获得在实例注册期间被阻塞的连接错误。该方法对从备份复制或者从活动数据库复制均有效。

DB_UNIQUE_NAME 初始化参数的影响

在 11g 中，如果存在于环境中的数据库与其他数据库有相同的 DB_NAME 参数，则必须设置 DB_UNIQUE_NAME 参数。例如，当报告的数据库与产品数据库处于同样的环境中，或当一个主数据库具有备用数据库，则企业中每个数据库的 DB_UNIQUE_NAME 的值必须具有唯一性。

正如我们指出的那样，RMAN 需要访问恢复目录以便从备用数据库中完成备份。为辅助该工作的实现，RMAN 能够感知恢复目录中 DB_UNIQUE_NAME 的设置，如果该参数被设置，则 RMAN 可以跟踪获取备份的准确位置，并且不会限制将其还原到何处(与

基于 DBID 的方法类似)。

例如，在同一环境下，在整个方案中应用两个备用数据库：备用数据库 1 有 5 分钟的延迟，备用数据库 2 有 30 分钟的延迟，这样能够确保当备用数据库 1 产生损坏时不会将损坏传播给备用数据库 2。在此环境下，应该分别将每个数据库的 DB_UNIQUE_NAME 设置到备用数据库 1 和备份数据库 2。RMAN 可以连接到任何需要备份的数据库，RMAN 知道备份来自于哪个数据库。记住在从备用数据库 1 备份时，若其在加载时间出错，在备用数据库 2 上执行的恢复仍然可运行，可以继续备份并且不会影响产品数据库。

20.2.1　从备用数据库生成的数据文件备份

从备用数据库备份数据文件是很常见的操作。执行下面的步骤可以从备用数据库生成数据文件的备份，然后在主数据库上使用这些备份：

(1) 从 RMAN 连接作为目标数据库的备用数据库，并连接到恢复目录：

```
RMAN> connect target sys/pswd@stby
RMAN> connect catalog rman/rman@rcat
```

(2) 生成数据库的备份。一旦执行 backup 命令，恢复目录就会自动执行再同步操作。

```
RMAN>backup database;
```

(3) 退出 RMAN。再次启动 RMAN(只有这样才能断开目标数据库和恢复目录的连接)：

```
RMAN> exit
$ rman
```

(4) 连接作为目标数据库的主数据库，并且连接到恢复目录：

```
RMAN> connect target sys/pswd@sun92
RMAN>resync catalog;
```

(5) 利用从备用数据库生成的备份执行 restore 操作：

```
RMAN> restore datafile 3;
RMAN> recover datafile 3;
```

20.2.2　从备用数据库备份归档日志

从备用数据库备份归档日志有些棘手，这是因为 RMAN 要查看 V$ARCHIVED_LOG 视图来确定要备份的归档日志。在主数据库上，在 LOG_ARCHIVE_DEST 中成功创建每个新归档日志后，V$ARCHIVED_LOG 视图的内容都会更新。但是在备用数据库上，只有备用数据库处于托管恢复模式(MANAGED RECOVERY)时(在这种模式中，备用数据库会自动应用归档日志)，V$ARCHIVED_LOG 视图才会更新。如果备用数据库不处于托管恢复模式，或者由于设置原因在备用数据库上会定期得到归档日志间隙，就很难成功地从备用数据库备份所有必需的归档日志。在这种情况下，建议您从主数据库备份归档

日志，而从备用数据库只备份数据文件。

20.2.3 使用闪回数据库重新实例化备用数据库

这里将介绍多个不同的 Oracle 技术，因为我们认为读者在阅读本章后可能会对 Data Guard 的实现感兴趣，但是如果您已阅读第 13 章的闪回技术，您可能不会很有兴趣。

在 Data Guard 实现中，一个最艰苦的时刻是将产品数据库故障恢复到备用数据库。即使进行了所有的准备和测试工作，这种故障恢复操作仍然会无法顺利完成。如果遇到这种情况，我们希望您能顺利解决问题。

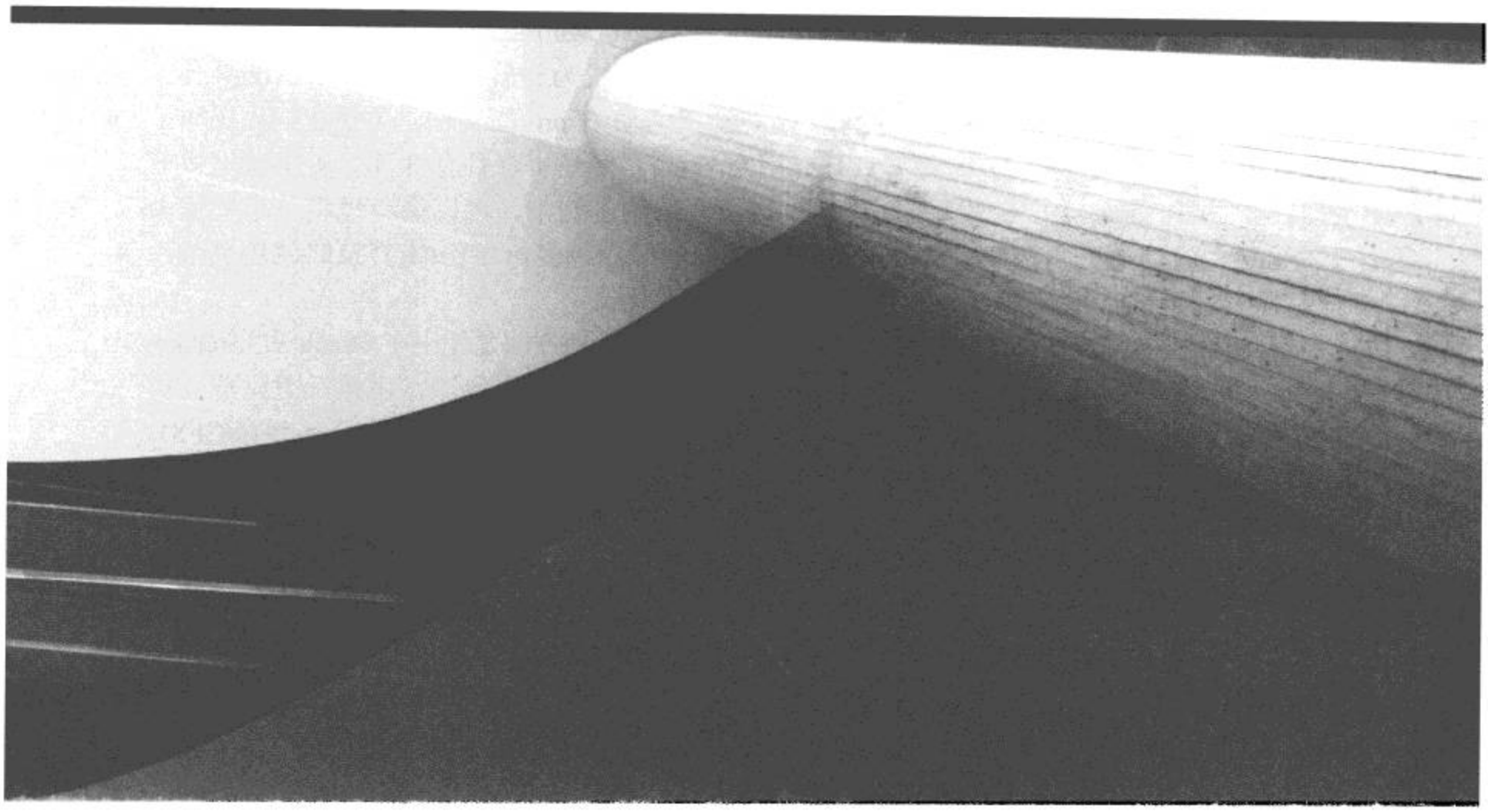
第二个最艰苦的时刻是决定如何推迟进行 Data Guard 部署。换句话说，现在备用数据库成为主数据库，接下来应该做什么？这实际上是另一本书要介绍的内容，但在此处有必要提出，因为 Oracle 使用自己的一些技巧来帮助处理接下来的艰苦步骤：在故障恢复后重新构建新的备用数据库。

下面是在 Oracle 9*i* 中进行的操作：假设产品数据库为 PROD，备用数据库为 PROD2。我们出于某些原因丢失了 PROD，因此故障恢复到 PROD2，PROD2 成为新的主数据库。然后，获得 PROD2 的全新备份，将其移动到驻留 PROD 的计算机，并且放置在 PROD 的顶部。接下来，我们开始将日志从 PROD2 移动到 PROD。在某些时刻，我们或许会有目的地将故障恢复到 PROD，然后重复这种大工作量和资源量的进程。

然而，在 Oracle Database 11*g* 中引入了新的重新构建备用数据库的方法：可以使用闪回数据库重新实例化备用数据库。当进行从主数据库到备用数据库的故障恢复时，过去主要是使用备份将新主数据库的副本还原到原有的主数据库，并且将其作为备用数据库，然后进行故障恢复。现在，我们可以使用原有的主数据库上的闪回数据库，将其移动到确切的 SCN。注意，这里未启用备用数据库。然后，我们建立从新的主数据库到原有的主数据库的归档日志流，向其提供新的备用控制文件，于是就可以很快恢复正常的业务工作。

20.3 小结

本章讨论了 RMAN 与备用数据库体系结构的关系。使用 RMAN 的 duplicate...for standby 命令可以创建备用数据库。我们讨论了使用 duplicate 命令创建备用数据库的方法，并且通过一个 RMAN 实习介绍了在主数据库所在的相同服务器上创建备用数据库的步骤。此外，我们还讨论了如何从已有的备用数据库创建备份以及 RMAN 如何使用这些备份在主数据库上执行还原操作，从而脱离产品环境进行工作。



第21章

RMAN与Real Application Clusters

虽然介绍Oracle Real Application Clusters(RAC)的复杂内容超出了本书的范围，但是我们可以提供为备份和恢复准备RAC配置的一些指导原则。和第20章中介绍的Data Guard一样，我们假设您已经实际使用过Oracle Database 11g中的RAC，因此只会作为提示而简要讨论RAC体系结构。

在贯穿本章的示例集群数据库中只具有两个节点：winrac1 和 winrac2。这两个节点共享一个磁盘阵列，使用 Oracle ASM 将该磁盘阵列配置为卷管理器。每个节点都具有一个实例：winrac1 节点具有 prod1 实例，winrac2 节点具有 prod2 实例。虽然我们只说明了 RAC 环境中最简单的情况(即集群只具有两个节点)，但是集群具有 3 个、4 个或更多节点时，其工作原理与工作方式是一样的。在所介绍的示例中，可以简单地将节点数量由 2 变为 3，将信道数由两个变为 3 个等。集群中的节点越多，备份/恢复策略就越复杂，但是需要应用的基本规则不会随实例数的变化而变化。

有两种跨多台计算机共享一个文件系统的方法。第一种方法是使用在 Oracle 9*i* Database 中引入的集群文件系统(Cluster File System，CFS)。集群文件系统通常由第三方供应商提供，例如 VERITAS；在 Windows 和 Linux 上，Oracle 提供了自己的集群文件系统(OCFS)。集群文件系统是根据集群对同时来自于多个节点的文件请求的正确处理和排队能力定义的——如果要实现集群的数据库，这种能力是必须具备的条件。

跨越多台计算机共享一个文件系统的另一种方法是使用在 Oracle Database 10g 中引入的自动存储管理(Automatic Storage Management，ASM)。ASM 是 Oracle 的第一个卷管理产品，它可以用于管理没有其他任何格式的原始磁盘。由于其体系结构，ASM 针对集群配置进行了构建，并且可以很容易地处理 RAC 的需求。我们可以认为 ASM 是针对 RAC 进行构建的，因为如果决定在 Oracle 标准版本上部署 RAC，ASM 就是共享的 Oracle 数据库文件所允许的唯一文件系统。

以上就是本书要介绍的所有内容，但是仍然需要说明的是：我们仍然可以在共享磁盘阵列上手动构建原始分区以安排 RAC 数据文件和日志文件。因此，可以根据情况而不需要集群文件系统或 ASM，只需要原始的、未加工的、共享的磁盘。然而，我们不推荐这种方法，同时也不会花费大量时间来解释如何配置 RMAN 来备份运行在原始磁盘上的 RAC 数据库。这是因为如果有机会使用 OCFS 或 ASM 来简化操作，那么就没有必要手动维护原始磁盘。特别是，当在 RAC 上执行备份和恢复操作时，使用 OCFS 和 ASM 将节省大量工作。而使用原始磁盘将只会带来更多复杂的工作。

21.1 Real Application Clusters: 独特的备份问题

您有必要先了解 RAC 集群的体系结构性质。实质上，集群至少包括两个不同的服务器，每个服务器都具有自己的内存和本地磁盘，并且都连接到一个共享的磁盘阵列。Oracle 通过创建两个实例来使用这个体系结构，每个节点上有一个实例，具有各自的 SGA/PGA 内存区域。每个实例都具有自己的重做日志，不过重做日志存在于共享的磁盘阵列上，并且能够被集群中的其他节点访问。所有控制文件和数据文件在两个实例之间共享，这意味着只有一个数据库但同时有两个线程被用来访问和更新这个数据库的数据。图 21-1 示例说明了最基本的 RAC。

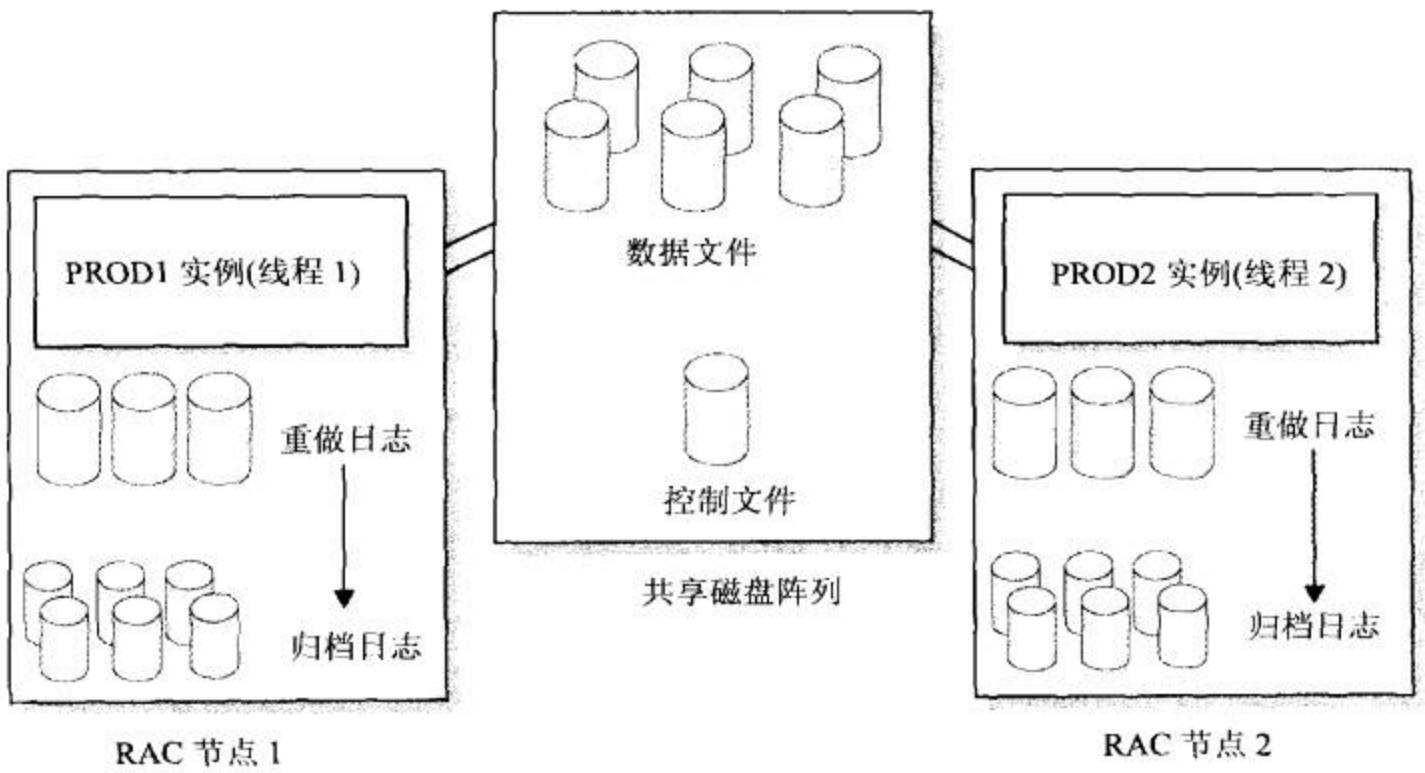


图 21-1　最基本的 RAC

RMAN 快照控制文件和 RAC

如果计划从多个节点运行备份，则需要将快照控制文件移到共享的位置。如果没有将快照控制文件移到共享的文件系统(例如 OCFS 或 ASM 磁盘组)，则必须确保本地目标在所有节点上都是相等的。使用 show 命令可以查看快照控制文件的位置：

```
rman> show snapshot controlfile name;
```

使用 configure 命令可以改变其中的值：

```
rman> configure snapshot controlfile name to
      '/u02/oradata/grid10/snap_grid10.scf';
```

从 RMAN 的角度看，这样的体系结构会在执行备份操作时产生一些需要注意的问题。首先，多个实例同时运行，但 RMAN 只能连接到一个节点，虽然在备份数据文件时不会产生问题，但是备份归档日志时就会产生一个问题：每个实例都会将自己的重做日志归档到本地驱动器，而不是归档到共享磁盘，这样的话，怎样才能获取其他归档日志呢？不过接下来还是先介绍数据文件备份。

21.1.1　数据文件备份

RAC 环境中的数据文件备份与单节点数据库中的数据文件备份非常相似：RMAN 连接到一个节点并执行 backup database 命令。RMAN 执行备份操作所需的内存是从这个节点中得到的。如果在磁盘上备份数据，备份就位于节点本地；如果在磁带上备份数据，就必须配置实例以集成 MML。

RAC 环境中数据文件备份的唯一问题是在单个节点上放置了过多的负荷。用户可以在单个节点上放置很多负荷，但如果不希望如此，还可以使用一种更好的方法。RMAN

最初只连接到单个节点，但是在实际的备份操作期间，RMAN 可以在所有节点上分配信道。下面的示例说明了这个操作：

```
configure default device type sbt;
configure device type sbt parallelism 2;
configure channel 1 device type sbt
connect 'sys/password@prod1';
configure channel 2 device type sbt
connect 'sys/password@prod2';
backup database;
```

随后，我们可以运行备份操作，同时 RMAN 在两个节点间展开工作。RAC 数据文件有时具有节点亲合力(node affinity)，此时从一个节点访问特定的数据文件会比从其他节点更快。如果集群中具有节点亲合力，RMAN 会得到相应的信息，并且能够从最快读取数据文件的节点备份这个数据文件。如果系统中不存在节点亲合力，RMAN 就只在两个信道间分发工作，这与使用两个信道并行化备份操作类似。显然，我们可以在每个节点上分配两个、三个或更多数目的信道。应该基于第 16 章中介绍的相同性能参数来决定每个节点上利用的信道数。

跨节点自动分布备份工作

在 Oracle Database 10*g* Release 2 中，RMAN 现在可以利用从 Oracle 的 Cluster Ready Services(CRS)中收集的信息来提供更好的 RAC 集成。最为重要的是，我们不再需要配置信道来在特定的节点上进行专门的连接。如果有两个节点并且设置并行化为 2，RMAN 就会在 CRS 中查询节点信息，并且自动在两个节点间扩展两个信道。此外，CRS 跟踪节点利用情况，并且将 RMAN 备份作业扩展到当前使用的这些节点上，从而避免 I/O 通信量堵塞。这是重要的自动化改进，我们可以让 CRS 自动完成更多的工作。可以让 CRS 确定如何在集群之间分布工作。只需要将并行化级别设置为备份中涉及的节点数，然后就可以让 CRS 完成工作。

21.1.2 归档日志备份

由于每个节点只负责自己的归档操作，也就是说每个节点都只能访问自己的非共享文件，因此归档日志备份比数据文件备份复杂许多。如果只连接到一个节点并执行 backup archivelog all 命令，RMAN 就在控制文件中查找来自所有节点的归档日志的列表。但是当查看本地节点时，RMAN 将只显示来自该节点的归档日志并显示出错。

可以将归档日志写入共享磁盘阵列上的原始分区吗？答案是：如果不介意出现管理问题，用户可以在共享磁盘阵列上的原始分区中写入归档日志。但使用原始分区时，在每个分区中只能写入一个文件，也就是说我们必须事先知道所有归档日志文件的名称并且创建到每个归档日志所在分区的符号链接。这会带来很多管理上的工作。更好的方法是使用具有集群文件系统或由 Oracle ASM 管理的共享磁盘卷。

如果坚持在每个节点上本地保留归档日志，则有一种可用的解决方案，该解决方案允许 RMAN 妥善处理非共享的磁盘位置。首先，确保每个节点都归档到一个唯一的文

件位置。例如，prod1 实例将日志归档到/u04/prod1/arch 目录，而 prod2 实例将日志归档到/u04/prod2/arch 目录。随后，我们可以在每个节点上分配信道，并且如前面介绍的数据文件备份的负载均衡那样备份归档日志，如下所示：

```
configure default device type sbt;
configure device type sbt parallelism 2;
configure channel 1 device type sbt
connect 'sys/password@prod1';
configure channel 2 device type sbt
connect 'sys/password@prod2';
backup archivelog all delete input;
```

RMAN 具有称为自动定位(autolocate)的功能，这个功能标识归档日志所属的节点，并且只从该节点尝试备份这些归档日志。通过这种方式，我们就不必在 RMAN 中指定需要备份的归档日志位于哪个节点上(RMAN 可以确定这些备份的日志)。

允许从单个节点执行归档日志备份操作的另一个方法是 NFS 加载其他节点的归档日志目标。例如，节点 winrac1 上的本地归档日志存储在/u04/prod1/arch 目录中，在这个节点上，我们可以 NFS 加载节点 winrac2 上的/u04/prod2/arch 驱动器为/u04/prod2/arch。这样，运行归档日志备份时，RMAN 在控制文件中检查归档日志位置，虽然只连接到 prod1 实例，但是 RMAN 会发现两个归档日志位置。图 21-2 示例了这种方法。

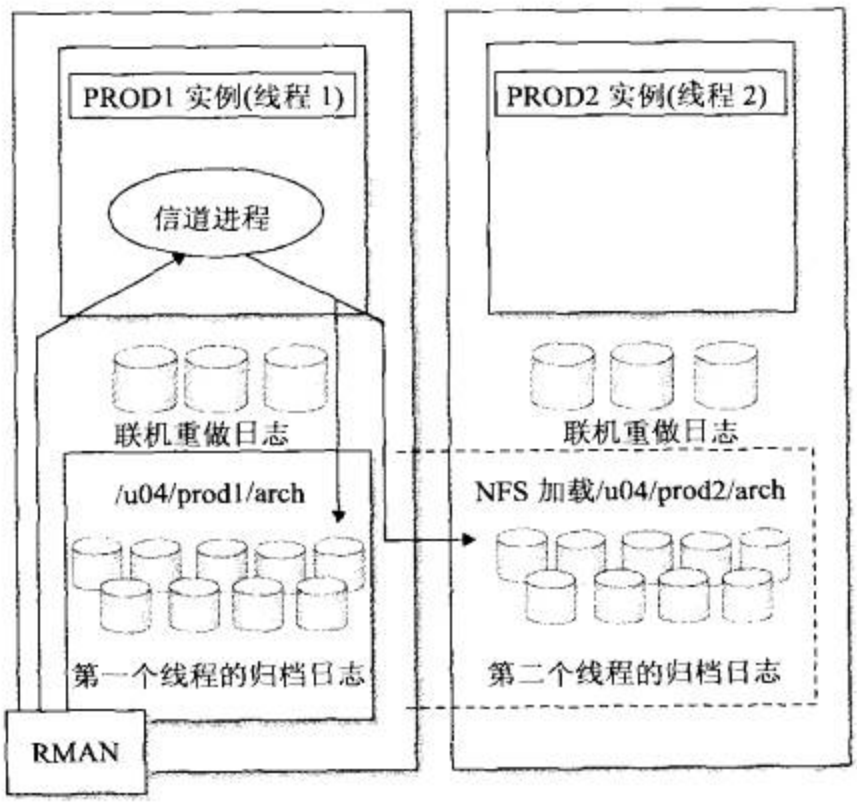


图 21-2　加载归档日志目标

在前面所提供的这些方法中出现的唯一问题是会得到一个单一的归档日志故障点。如果只将日志归档在各自的节点上，在丢失某节点时就会丢失这个节点上的归档日志。也就是说我们必须执行整个数据库的时间点恢复，将数据库恢复到丢失该节点前的时间点。

更好的策略是设置每个节点都使用 LOG_ARCHIVE_DEST_2 参数，在另一个节点上

写入归档日志。本章前面讨论的 NFS 加载策略就是实现这种任务的一种方法。但不是在只读模式中 NFS 加载其他节点上的归档目标，而是在写模式下 NFS 加载其他节点上的驱动器，并设置这个 NFS 加载为第二个归档目标。以前面的两节点 RAC 数据库为例，在 winrac1 节点上可以加载 winrac2 节点上共享的/u04/prod2/arch 目录，而在 winrac2 节点上可以加载 winrac1 节点上共享的/u04/prod1/arch 目录，然后为每个节点创建如下所示的 init.ora 文件：

```
winrac1 init.ora file:
    log_archive_dest_1='location=/u04/prod1/arch'
    log_archive_dest_2='location=/u04/prod2/arch'
    ...
    winrac2 init.ora file:
    log_archive_dest_1='location=/u04/prod2/arch'
    log_archive_dest_2='location=/u04/prod1/arch'
```

此时，Oracle 可以将归档日志从每个节点写入另一个节点的归档目的地。这种解决方案能够从单个节点备份归档日志，在节点丢失时能提供容错。

使用 ASMT 和 FRA 避免归档日志备份的复杂情况

通过在共享磁盘阵列上创建一个位置可以避免所有这些复杂的情况，且该磁盘阵列上具有经过集群文件系统(例如 OCFS)加工的卷。更好的方法是，使用 ASM 作为卷管理器对所有 RAC 文件进行管理。然后，我们可以创建另一个磁盘组，并且将闪回恢复区(FRA)用于该磁盘组。通过在 ASM 磁盘组上使用 FRA，在 RAC 中管理归档日志就如同在单线程的环境中管理这些归档日志。如果配置数据库归档到 FRA，两个节点将在一个它们都可见的位置中存放日志。在清除操作期间管理归档日志等同于在单线程的数据库中管理这些归档日志。

RMAN、RAC 和 Net 连接

RAC 提供许多强大的负载均衡和故障切换功能作为 Net 配置的一部分，但是，这意味着要在 listener.ora 文件和 tnsnames.ora 文件中针对集群节点和客户端做相应的修改。不过，RMAN 对这些功能的要求非常严格。RMAN 只能连接到一个节点，并且不能实现故障切换或负载均衡。因此，用于目标数据库连接和信道分配 connect 子句的 Net 别名必须被配置为连接到具有专用服务器的一个节点。然而，在 Oracle Database 11g 中，我们不能使用与其他连接的故障切换所配置的相同的 NET 别名。

21.2 RAC 恢复问题

在 RAC 环境中执行还原与恢复操作比执行备份操作更为复杂，这同样是由于一个具有多个实例的数据库的复杂特性所引起的。相关的问题可以大致分为 3 类：还原操作、介质管理问题和恢复问题。

21.2.1　还原操作

我们必须再次指出：从 RMAN 执行还原操作时，RMAN 只连接到一个节点，不过此后可以在每个节点上分配信道。要做到这一点并不难，真正困难的是跟踪文件是从哪个节点备份的。

文件可访问性是保证在 RAC 节点上成功执行还原操作的关键。如果使用每个站点上所分配的信道在磁盘上备份数据，就必须在还原操作期间在每个节点分配相同的信道。我们知道，如果节点没有丢失，这样操作是没问题的。但如果节点出现故障，就不能访问这个节点上的磁盘备份，还原操作也会失败。教训在于，如果将这么多时间和财力用在 RAC 上以确保不出现单一的故障点，那么要确保在备份与恢复策略中也贯彻这种思想。如果要在磁盘上备份数据，就需要在多个节点上复制备份，这样可能意味着要重写默认的自动定位功能，以便能够指定从哪里备份文件。如下所示，通过为数据文件集指定信道，我们可以完成这个操作：

```
configure default device type disk;
configure device type sbt parallelism 2;
configure channel 1 device type disk
connect 'sys/password@prod1';
configure channel 2 device type disk
connect 'sys/password@prod2';
backup (datafile 1,2,3,4,5 channel ORA_DISK_1)
       (datafile 6,7,8,9,10 channel ORA_DISK_2);
# then switch the channels for the datafiles
backup (datafile 1,2,3,4,5 channel ORA_DISK_2)
       (datafile 6,7,8,9,10 channel ORA_DISK_1);
```

同样，使用 ASM 可以节省大量时间和工作。更为重要的是，在共享的 ASM 磁盘卷上运行 FRA 将简化 RAC 环境中的磁盘备份工作，并且除了单线程数据库中的操作之外只需要很少的配置工作。在 ASM 上使用 FRA 可以利用节点之间的作业多路复用，同时所有备份最终仍然在一个位置中，还原操作期间的其他节点都可以访问该位置。因此，我们不再需要担心磁盘备份会在停机的节点上停止而无法访问。也不需要担心单一的故障点，因为 ASM 提供了自动冗余(automatic redundancy)，我们可以镜像 FRA，并且具有备份的双向副本，而不需要任何额外的工作。

21.2.2　还原操作期间的介质管理问题

为了使还原操作期间的所有节点都可使用备份，另一种更好的方法是在磁带上备份数据，并且使用集中化的介质管理服务器保存磁带备份。如果每个节点上都具有磁带设备，并且这些磁带都用于磁带备份操作，就会不必要地增加复杂度。如果丢失了集群中的一个节点，就会丢失这个节点生成的所有备份的介质管理目录。虽然您的介质管理产品可能有自动的目录备份，但是您现在要执行另一次还原操作，那还原到哪里呢？何时执行？此时，需要一个包含其他节点备份信息的目录。虽然灾难能够被恢复，但是我们必须手动执行所有操作，这样就会花费大量的时间。

我们推荐您使用集中化的介质管理系统，这样所有节点都能在由同一个介质管理服务器管理的磁带设备上备份数据，此时只存在单个介质管理目录。在这种情况中，如果丢失了一个节点，就可以简单地在 RMAN 信道中指定这个客户端的名称，并且还原到不同的节点上。使用 RMAN 从磁带还原备份时，切记：执行还原操作时必须考虑 RMAN 生成备份所在的节点。

例如，在前面介绍的两节点集群中，假设丢失了节点 winrac2，由此引发的灾难会使得某些数据文件发生损坏。因为已经使用了在每个节点上分配信道执行备份操作的备份策略，所以还原操作也必须在每个节点上分配信道。然而，此时没有节点 winrac2，我们只能在节点 winrac1 上分配两种信道：常规的磁带信道与指定客户端为 winrac2 的信道，如下所示：

```
configure default device type sbt;
configure device type sbt parallelism 2;
configure channel 1 device type sbt
parms="env=(nb_ora_serv=rmsrv)";
configure channel 2 device type disk
parms="env=(nb_ora_serv=rmsrv, nb_ora_client=winrac2)";
restore datafile 5,12,13,18;
```

这个示例非常简单地解决了一个复杂的问题，但执行完上述的所有操作后，我们建议您使用只从集群中的单个节点备份数据文件的备份方法。如果由于数据库大小的原因而不能使用这种简单的方法，可以尝试将 RMAN 备份限制为节点较小的子集。这样，我们可以降低还原操作的复杂程度。

另一种方法是利用 Oracle Database 10*g* 中的新功能，首先将备份放置到一个磁盘位置(使用 ASM 磁盘卷上的一个 FRA)。如果在磁盘阵列上具备放置备份的磁盘空间，则备份会在一个控制的时间内从一个或多个节点最终移动到磁带。使用 backup recovery area 命令将可以从 FRA 获得统一的备份集，并且通过一个操作将它们全部移动到磁带。可以从任何节点或任何节点组合中执行这种磁带操作。如果从一个节点中执行磁带备份，则在还原操作期间可简化 SBT 信道参数，即使使用的是不同的节点执行该还原操作。

21.2.3 还原操作后的恢复问题

还原文件后，就可以通过应用归档日志来执行介质恢复操作。在 RAC 环境中执行介质恢复操作必须记住：只有一个节点可以执行恢复操作。应当牢记这一点。这意味着一个节点必须能够访问磁盘上的所有归档日志。因此，如果使用了每个节点在一个本地磁盘中保存各自归档日志的归档日志策略，就必须使恢复节点能够使用这个本地磁盘。如果遵守了本章前面 21.1.2 小节中介绍的原则，就可以通过 NFS 完成介质恢复操作。我们可以简单地加载其他节点的归档日志目标，并且从 RMAN 中执行恢复命令。如果要使用 CFS，这不会成为问题，此时可以忽略归档日志节点亲合力问题。如果在 ASM 上使用 FRA，则访问所有归档日志也不会成为问题。

如果需要从 RMAN 备份中还原归档日志，还原归档日志与还原数据文件所应用的规则和原则是相同的。如果在每个节点都创建用于备份操作的信道，在还原操作时也要创

建相同的信道。如果丢失了一个节点，就必须分配一个包含该客户端名称的信道，以便介质管理器查找适当的备份(参阅本章前面的 21.2.2 小节)。此外，如果 LOG_ARCHIVE_DEST 参数是在丢失的节点上，而不是在执行还原操作的节点上，可能还必须将归档日志还原到本地的一个目录中，如下所示：

```
restore archivelog like '%prod2%' to '/u04/prod1/arch%';
```

使用 RMAN 时，虽然只有一个节点可以执行恢复，但 RMAN 可以并行执行介质恢复。我们不能控制这种并行化级别，唯一的操作是关闭它：

```
rman> Recover database noparallel;
```

然而，RMAN 在默认情况下基于执行恢复的节点上的可用 CPU 数量来自动选择恢复的并行度。

21.3　高级的 RMAN/RAC 功能

一旦确定了 RAC 数据库所使用的备份与恢复策略，在 RAC 环境中就能利用 RMAN 在单节点数据库环境中提供的许多优点：块损坏检查、空值压缩以及块介质恢复。此外，RAC 环境中也存在一些高级功能(如数据库复制)。RAC 数据库的 RMAN 备份可用于复制操作和备用数据库的创建，这与单节点系统的用法相同。然而，有一些注意事项要注意，下面对此进行介绍。

21.3.1　复制到单节点系统

作为 RAC 集群的管理员，如果不确定 RMAN 是适用的工具，可参照以下功能：RAC 数据库的 RMAN 备份能够用于在单节点数据库上创建 RAC 数据库的副本。这样，不必购买和管理额外的 RAC 集群就能够得到产品数据库的副本。此时，我们可以在已加工的文件系统上运行一个单节点数据库。

实际上，RMAN 不可以从一个 RAC 集群复制到另一个 RAC 集群，它只可以复制到单线程的数据库。然而，一旦进行复制操作，我们就可以很容易地将副本数据库转换为 RAC 数据库，只要确保复制到安装 CRS 的节点上的 ASM 磁盘组即可。

RMAN 实习：将一个 RAC 数据库复制到单节点系统

实习注意事项

本实习将创建两节点数据库的单节点复制，这个复制既可以创建在新的服务器上，也可以创建在 RAC 集群中某个节点上的已加工文件系统中。在这个示例中，我们将在 RAC 集群中某个节点上的文件系统中复制 RAC 数据库。由于复制必须完成恢复操作，而且恢复会话只能访问执行恢复操作的节点，因此这个节点必须能够访问所有节点的归档日志。本实习假定在每个节点上 NFS 加载了其他节点上的归档日志目标，这样每个归档日志流的完全副本在每个节点上都有效。

集群中的两个节点是 opcbs01 和 opcbs02，其各自的实例是 V112A 和 V112B。执行所有 RMAN 操作时连接的是 V112B 实例。

步骤 1 构建辅助数据库目录结构:

```
mkdir /u02/32bit/app/oracle/oradata/aux1
mkdir /u02/32bit/app/oracle/oradata/aux1/arch
cd /u02/32bit/app/oracle/admin
mkdir aux1
cd aux1
mkdir pfile bdump udump cdump
ls
```

步骤 2 将目标数据库的 init.ora 文件复制到辅助数据库的相应位置上。如果目标数据库使用了服务器参数文件(SPFILE)，为了获取要移动的参数，就必须从这个服务器参数文件(SPFILE)中创建一个参数文件(PFILE)。

如果在目标数据库上使用了服务器参数文件，就输入下面的命令:

```
SQL> connect / as sysdba
create pfile='/u02/32bit/app/oracle/admin/aux1/pfile/init.ora'
from spfile;
```

如果在目标数据库上使用了 init.ora 文件，就输入下面的命令:

```
cp /u02/32bit/app/oracle/admin/V112B/pfile/init.ora
   /u02/32bit/app/oracle/admin/aux1/pfile/init.ora
```

步骤 3 根据需要完成对 aux1 实例的 init.ora 文件的所有修改:

```
control_files=
   '/u02/32bit/app/oracle/oradata/aux1/control01.dbf'
core_dump_dest='/u02/32bit/app/oracle/admin/aux1/cdump'
background_dump_dest='/u02/32bit/app/oracle/admin/aux1/bdump'
user_dump_dest=/u02/32bit/app/oracle/admin/aux1/udump
log_archive_dest_1=
   'location=/u02/32bit/app/oracle/oradata/aux1/arch'
db_name='aux1'
instance_name='aux1'
remote_login_passwordfile=exclusive
db_file_name_convert=
('/dev/vx/rdsk/usupport_dg', '/u02/32bit/app/oracle/oradata/aux1')
```

可以完全删除下面的参数(包括引用其他实例的那些参数):

```
cluster_database_instances=2
cluster_database=true
V112A.instance_name='V112A'
V112B.instance_name='V112B'
V112B.instance_number=2
V112A.instance_number=1
V112B.thread=2
```

```
V112A.thread=1
V112B.undo_tablespace='UNDOTBS2'
V112A.undo_tablespace='UNDOTBS1'
```

我们可以只使用下面的参数来代替:

```
undo_tablespace='UNDOTBS2'
```

步骤 4　为 aux1 实例创建口令文件。参阅第 19 章中的 19.2.4 小节的 RMAN 实习。

步骤 5　以未加载模式启动 aux1 实例:

```
ORACLE_SID=aux1
export ORACLE_SID
SQLplus /nolog
SQL>connect / as sysdba
SQL>startup nomount
pfile=/u02/32bit/app/oracle/admin/aux1/pfile/init.ora
```

步骤 6　配置连接 aux1 实例的网络文件。修改 listener.ora 文件后，确保重新运行侦听程序，否则修改不会起作用:

```
lsnrctl
LSNRCTL>stop
LSNRCTL>start
```

tnsnames.ora 文件应当具有下面的条目:

```
AUX1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = opcbsol2)(PORT = 1526))
    )
    (CONNECT_DATA =
      (SID = aux1)
      (SERVER = DEDICATED)
    )
)
```

listener.ora 文件应当具有下面的条目:

```
(SID_DESC =
  (GLOBAL_DBNAME = aux1)
  (ORACLE_HOME = /u02/32bit/app/oracle/product/11.2.0/dbhome_1)
  (SID_NAME = aux1)
)
```

步骤 7　从 RMAN 连接到目标实例和辅助实例，并且运行 duplicate 命令:

```
ORACLE_SID=V112B
export ORACLE_SID
rman
```

```
rman>connect target /
rman> connect auxiliary sys/password@aux1
rman>duplicate target database to aux1
  pfile=/u02/32bit/app/oracle/admin/aux1/pfile/init.ora
  logfile
  '/u02/32bit/app/oracle/oradata/aux1/redo1.dbf' size 100m,
  '/u02/32bit/app/oracle/oradata/aux1/redo2.dbf' size 100m,
  '/u02/32bit/app/oracle/oradata/aux1/redo3.dbf' size 100m;
```

21.3.2 单节点上的备用数据库

当然，如果可以在单节点上复制数据库，也可以使用 duplicate 命令在单个节点上为 RAC 集群创建备用数据库。除了更加直接的复制操作外，这个功能还为用户提供一个优秀的成本性能策略，该策略为 RAC 数据库提供了一个灾难恢复解决方案。我们不必创建另一个要添加硬件和软件的完整的 RAC 系统(在备用数据库中不会用到这些硬件和软件)，而只需要在单节点系统上创建备用数据库。显然，这样的数据库不具有 RAC 数据库的计算能力或负载平衡功能，但是却可以提供一个合理的灾难解决方案，以保证能够还原 RAC 数据库。

和复制进程相同，关键在于 DB_FILE_NAME_CONVERT 参数，这个参数将文件位置从 OCFS 或 ASM 磁盘组切换到普通的、未集群的文件系统。此外，单节点备用数据库可以从 RAC 集群的每个节点中接收归档日志，并以正确的时间顺序应用这些归档日志。

RMAN 实习：创建 RAC 数据库的单节点备用数据库

步骤 1 使用 RMAN 创建备用控制文件：

```
ORACLE_SID=V112B
export ORACLE_SID
rman
rman> connect target /
rman> backup current controlfile for standby
    format= '/u02/backup/stby_cfile.%U';
```

创建完这个备用控制文件后，需要指定一个时间点，以完成一些日志切换，然后记录 V$ARCHIVED_LOG 视图中的最后一个日志序列号。因为下面的命令能够强制在所有节点上完成日志切换，所以选择哪个线程无关紧要：

```
SQL> alter system archivelog current;
SQL> select sequence# from v$archived_log;
```

步骤 2 构建备用数据库目录结构：

```
mkdir /u02/32bit/app/oracle/oradata/stby
mkdir /u02/32bit/app/oracle/oradata/stby/arch
cd /u02/32bit/app/oracle/admin
mkdir stby
cd stby
mkdir pfile bdump udump cdump
ls
```

步骤 3　将目标数据库的 init.ora 文件复制到辅助数据库的相应位置上。如果目标数据库使用了服务器参数文件(SPFILE)，为了获取要移动的参数，就必须从这个服务器参数文件中创建一个参数文件(PFILE)。

如果在目标数据库上使用了服务器参数文件，就输入下面的命令:

```
SQL> connect / as sysdba
create pfile='/u02/32bit/app/oracle/admin/stby/pfile/init.ora'
from spfile;
```

如果在目标数据库上使用了 init.ora 文件，就输入下面的命令:

```
cp /u02/32bit/app/oracle/admin/V112/pfile/init.ora
   /u02/32bit/app/oracle/admin/stby/pfile/init.ora
```

步骤 4　根据需要完成对 stby 实例的 init.ora 文件的所有修改:

```
control_files= '/u02/32bit/app/oracle/oradata/stby/control01.dbf'
background_dump_dest=/u02/32bit/app/oracle/admin/stby/bdump
user_dump_dest=/u02/32bit/app/oracle/admin/stby/udump
log_archive_dest_1=
   'location=/u02/32bit/app/oracle/oradata/stby/arch'
standby_archive_dest=
   'location=/u02/32bit/app/oracle/oradata/stbyarch'
lock_name_space='stby'
remote_login_passwordfile=exclusive
db_file_name_convert=
  ('/dev/vx/rdsk/usupport_dg', '/u02/32bit/app/oracle/oradata/aux1')
log_file_name_convert=
('/dev/vx/rdsk/usupport_dg', '/u02/32bit/app/oracle/oradata/aux1')
```

可以完全删除下面的参数(包括引用其他实例的那些参数):

```
cluster_database_instances=2
cluster_database=true
V112A.instance_name='V112A'
V112B.instance_name='V112B'
V112B.instance_number=2
V112A.instance_number=1
V112B.thread=2
V112A.thread=1
V112B.undo_tablespace='UNDOTBS2'
V112A.undo_tablespace='UNDOTBS1'
```

我们可以只使用下面的参数来代替:

```
instance_name='V112B'
undo_tablespace='UNDOTBS2'
```

步骤 5　为 stby 实例创建口令文件。参阅第 19 章中 19.2.4 小节的 RMAN 实习。

步骤 6 以未加载模式启动 stby 实例:

```
ORACLE_SID=stby
export ORACLE_SID
SQLplus /nolog
SQL>connect / as sysdba
SQL>startup nomount
pfile=/u02/32bit/app/oracle/admin/stby/pfile/init.ora
```

步骤 7 配置连接 stby 实例的网络文件。修改 listener.ora 文件后，确保重新运行侦听程序，否则修改不会起作用:

```
lsnrctl
LSNRCTL>stop
LSNRCTL>start
```

tnsnames.ora 文件应当具有下面的条目:

```
STBY =
    (DESCRIPTION =
      (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = opcbsol2)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = stby)
      (SERVER = DEDICATED)
    )
)
```

listener.ora 文件应当具有下面的条目:

```
(SID_DESC =
  (GLOBAL_DBNAME = aux1)
  (ORACLE_HOME = /u02/32bit/app/oracle/product/11.2.0/dbhome_1)
  (SID_NAME = aux1)
)
```

步骤 8 从 RMAN 连接到目标实例和辅助实例，并且运行 duplicate 命令:

```
ORACLE_SID=V112B
export ORACLE_SID
rman
rman>connect target /
rman> connect auxiliary sys/password@stby
rman>run {
  set until sequence = 43 thread = 1;
  duplicate target database for standby
  dorecover;}
```

21.3.3　从单节点备用数据库备份多节点 RAC 数据库

一旦创建了单节点备用数据库，就可以像在普通环境中一样从这个备用数据库生成所有备份。也就是说将产品 RAC 备份操作从 RAC 集群移到创建和运行备用数据库的节点上。这样可以减轻集群的负载，为用户提供一个灾难恢复解决方案。并且由于来自所有节点的归档日志都肯定位于备用数据库中，所以还可以为用户提供一个简化的归档日志备份解决方案。

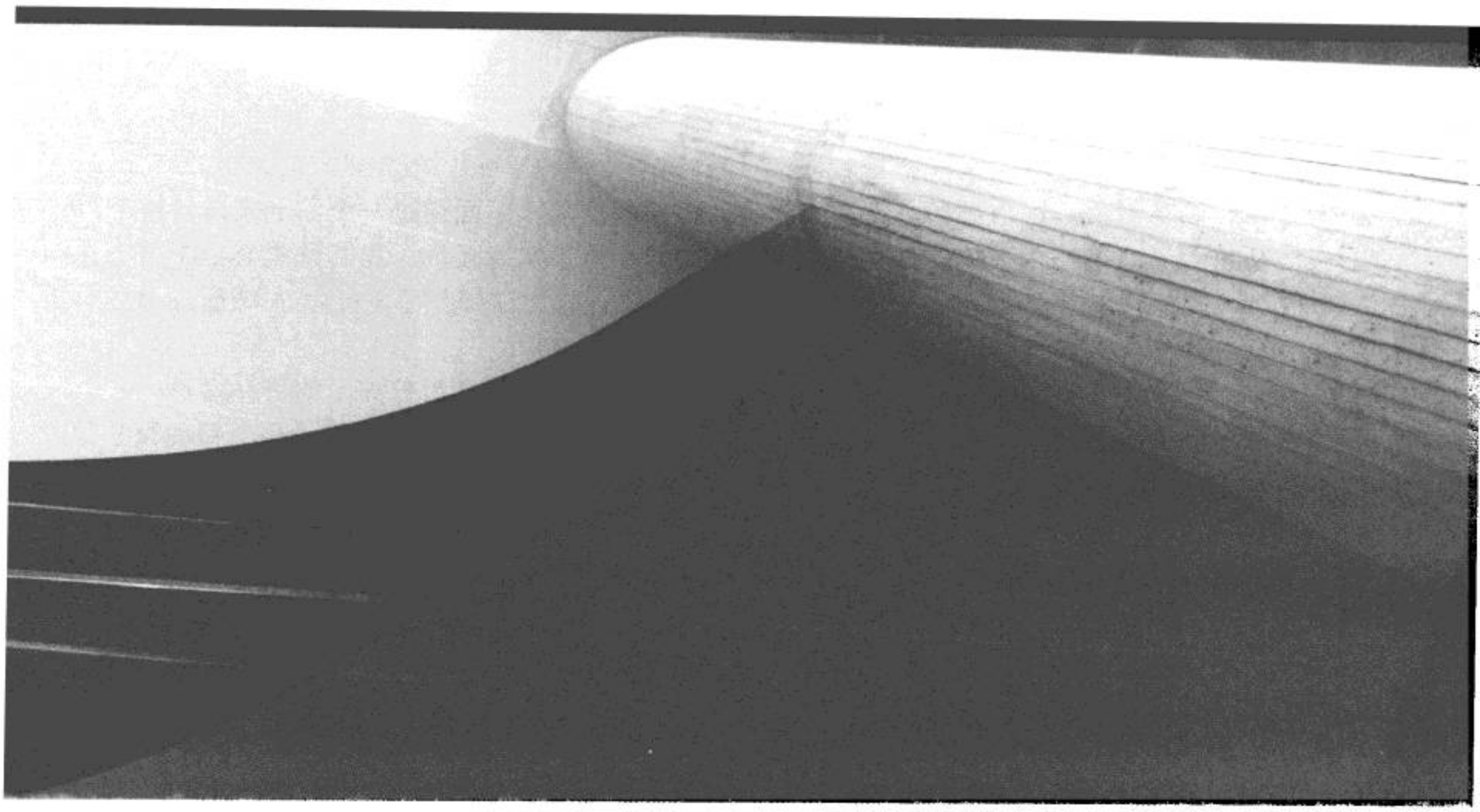
同样，这个操作的关键仍是 DB_FILE_NAME_CONVERT 参数。该备份是从其数据文件位于已加工文件系统上的备用数据库中获得，但备用数据库控制文件也记录了这些文件的原始位置(原始系统)。因此，在产品 RAC 数据库上还原从备用数据库生成的备份时，RMAN 会在控制文件中查找文件的原始位置，并且将文件还原到这些位置。

要应用上述的这个解决方案，就必须使用恢复目录。恢复目录作为转换代理程序，它将与备份有关的元数据从备用数据库控制文件转换到主数据库控制文件。从备用数据库生成备份后，RMAN 会与恢复目录再同步，记录备份元数据。随后，连接 RMAN 与主数据库并执行手动再同步操作时，来自备用数据库控制文件的备份元数据会被置入主数据库控制文件的记录中。

从备用数据库执行备份操作时，确保将备用数据库作为目标数据库连接是非常重要的。同样，在执行再同步操作时，将主数据库作为目标数据库连接也是非常重要的。RMAN 在执行操作时是看不到两者功能上的区别的：即它们的 DB_NAME、DBID 和重做流历史记录都相同。

21.4　小结

本章讨论了 RMAN 与 RAC 集群中的数据库交互的方法。介绍了 RMAN 如何能够在每个节点上分配用于备份操作的信道，不过恢复操作要求从单个节点可以访问所有备份。我们还讨论了由于不同节点上有多个重做线程而导致归档日志备份的复杂化。最后，本章介绍了一些 RMAN 实习(包括将 RAC 数据库复制到单节点数据库、从 RAC 数据库创建单节点备用数据库)。



第 22 章

同步和拆分技术中的 RMAN

同步和拆分技术就是存储恢复方面的创新解决方案的示例，该技术补充或复制 RMAN 可以独立完成的各项功能。过去 5 年来，同步和拆分已经成为广泛使用的技术，该技术提供了存储硬件层次上的直接而快速的系统恢复。

本章中将简要介绍同步和拆分技术。我们不会讨论特定的单一实现，而是讨论该技术对于 RMAN 和数据库备份的含义。简要介绍该技术之后，我们就会介绍将同步和拆分解决方案集成到 RMAN 备份策略中所需的特定步骤。

22.1 同步和拆分：不完整的镜像备份

执行同步和拆分备份最初只涉及扩展硬件镜像的功能，下面通过一个示例来进行解释。假设有一个包含两个硬盘驱动器的磁盘控制器。为了实现冗余，我们设置 RAID 级别为 0+1，从而将磁盘 A 中的所有内容镜像到磁盘 B。该方法提供了对磁盘 A 或磁盘 B 上任何硬件故障的直接保护。

然后，下一步是尝试利用硬件镜像来提供逻辑容错。同步和拆分技术的目标如下：若镜像中的两个副本出错，它能为之提供回退的位置。例如，用户已经删除了整个 Oracle 软件树或 oradata 目录的情况下。这种删除操作将直接作用于镜像中的两个副本，因此镜像的副本将没有任何实际的作用。

因此，解决方案是：任何镜像的磁盘组都有两个镜像组，但只有一个镜像是按主磁盘组的位写入。例如，我们构建一个具有 3 个逻辑卷(A、B 和 C)的示例，所有这些逻辑卷都专用于保存相同的数据。卷 A、B 和 C 互相作为镜像副本。然而，在下午 2 点时，拆分卷 A 和其镜像，并且在拆分时保持其数据位不变。卷 B 和卷 C 继续作为按位存储的副本。4 个小时后(下午 6 点)，拆分卷 C 和卷 B，从而不再对其写入数据。此时，在该卷上有数据的 3 个不同的副本：下午 2 点时的副本、下午 6 点时的副本，以及当前的副本，并且没有任何冗余来防止磁盘故障。

RAID 简介

关于 RAID 有数百种理论，从先进的理论到传统的理论，这些理论概括了磁盘故障保护的最佳可用解决方案。一般来说，在 Oracle 中使用的最佳解决方案是 RAID 0+1，其中有两个磁盘组，组 1 和组 2，每个组中都有两个磁盘。组 1 上的两个磁盘使用分条的方式，从而使数据均匀地分布到两个磁盘上。组 2 是组 1 的按位副本。这种配置提供了性能(通过在多个磁盘之间分条来避免热点)和冗余(通过将每一位写入两次)。

最新的理论是 RAID 1+0 配置，这种配置与 RAID 0+1 有细微的区别。不是先分条后进行镜像，RAID 1+0 配置是先进行镜像后分条。图 22-1 很好地显示了这种区别。在该图中，我们分别镜像每个磁盘，从而最终有 4 个磁盘组。镜像每个磁盘后，在 4 个镜像之间进行分条操作。

虽然看起来只有很小的区别，但 RAID 1+0 具有更好的容错性能，因为任何一个磁盘的故障都不会影响到另一个对应的镜像磁盘。在 RAID 0+1 中，如果组 2 中的任何磁盘失效，则整个组将脱机。因此，在多个磁盘故障而不是单个磁盘故障方面，RAID 1+0 提供了比 RAID 0+1 更好的容错性能，如图 22-1 所示。

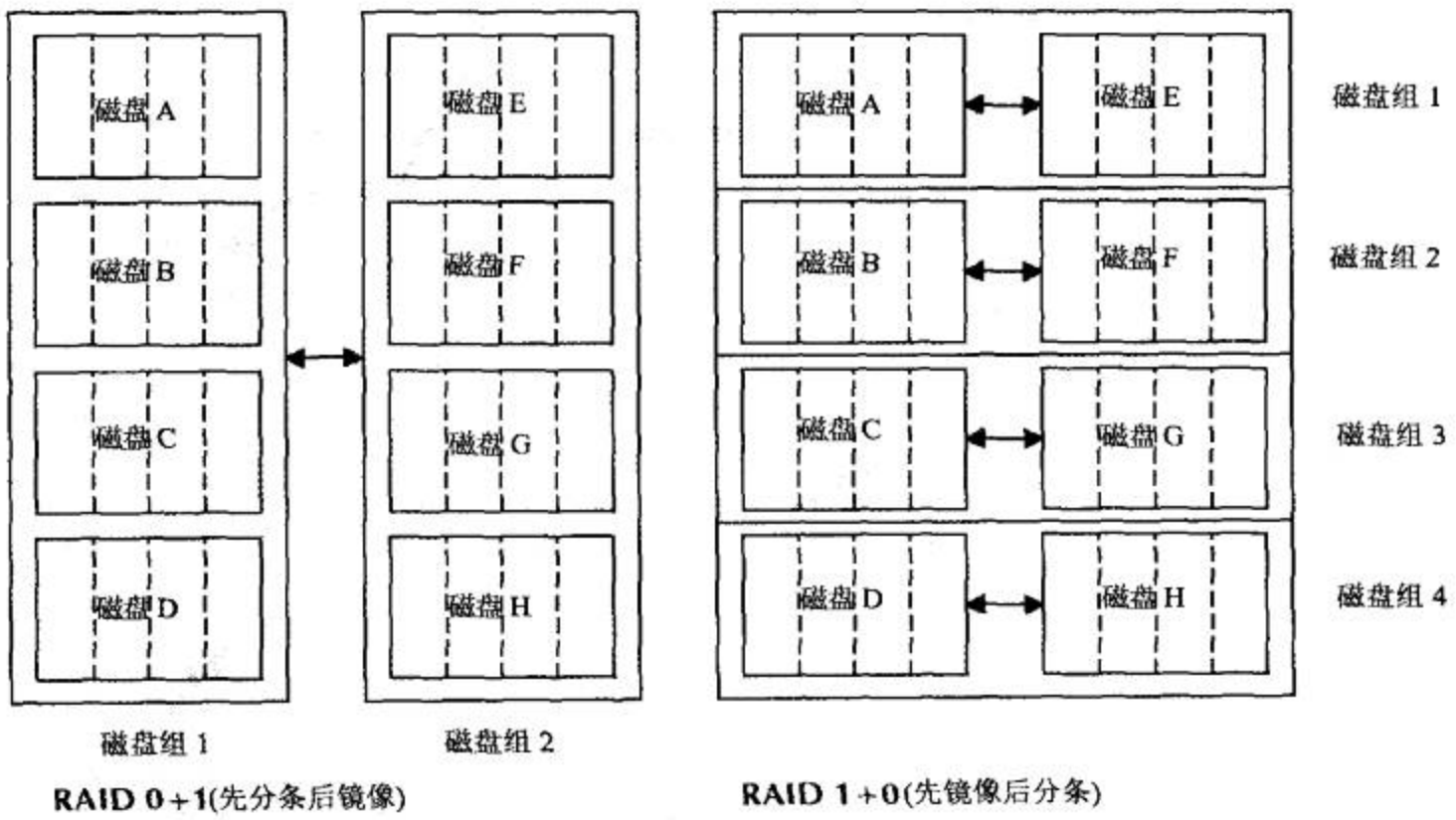


图 22-1　RAID 0+1 配置与 RAID 1+0 配置的区别

为了恢复到 RAID 0+1 配置，磁盘卷 A 将重新与当前正在运行的磁盘 B 同步。这种同步基于如下情况：这些卷有适当的日志记录机制，该机制记录所有的数据更改。这种日志记录除了每个卷的多个写入之外还需要更多的 I/O。卷 A 将访问卷 B 上的更改日志，并且应用所有更改，直到获得卷 B 的当前写入数据。此时，我们有冗余模式的卷 A 和卷 B，卷 C 则是下午 6 点时的回退位置。图 22-2 显示了该过程。

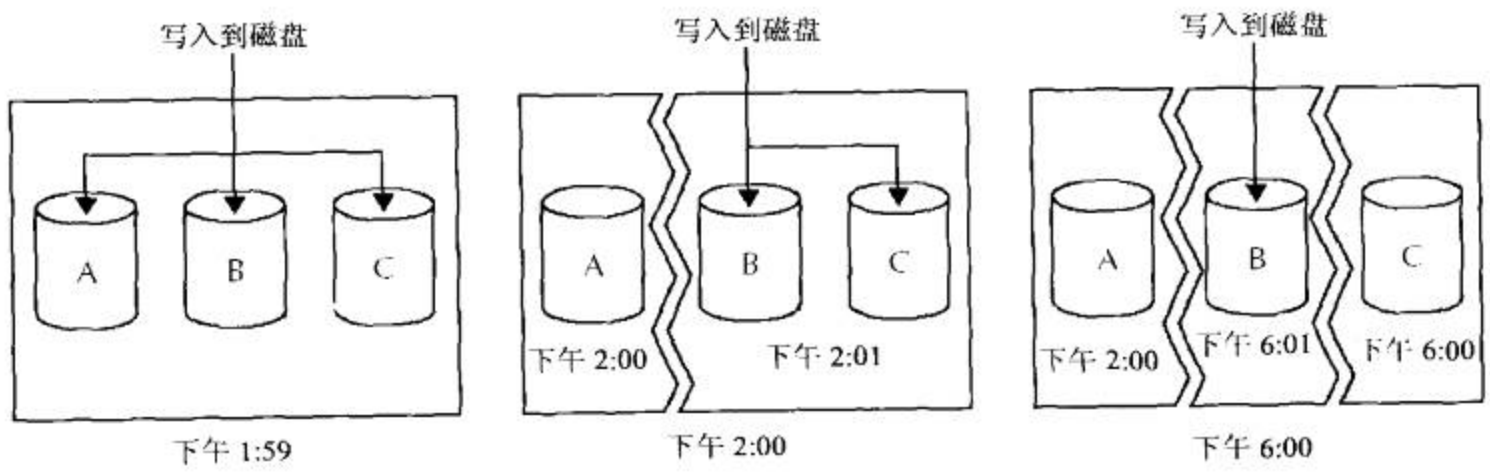


图 22-2　实际运行的同步和拆分技术

这种同步和拆分循环会持续进行。每隔 4 个小时，某个卷就与主卷同步，同时拆分另一个卷以提供发生逻辑故障时的回退位置。在遇到逻辑故障时会发生什么？在该示例中，假设当前时间是下午 8 点，卷 A 和卷 B 具有并发的写入，而卷 C 则在下午 6 点时空闲等待。下午 8 点时，DBA 执行了一些系统维护，并且从产品数据库中删除了系统数据文件，这时就会产生错误。

幸运的是，到当天的下午 5 点为止，并没有向该数据库添加任何不可恢复的数据。

然而，每晚的批处理加载工作将在大约 15 分钟之后开始。DBA 通过一个小窗口来备份和运行产品数据库。

当数据库完全在镜像磁盘卷 A 和 B 上运行时，同步和拆分体系结构就可以向 DBA 提供直接的解决方案。DBA 可以直接将在下午 6 点后保持不变的卷 C 配置为主卷，并且启动数据库。当数据库查找它的数据文件时，它会发现卷 C 上的所有文件都处于下午 6 点时的状态，没有发生任何删除操作。DBA 完成工作的时间是下午 8:05。批处理进程将按时取消。图 22-3 显示了该进程。

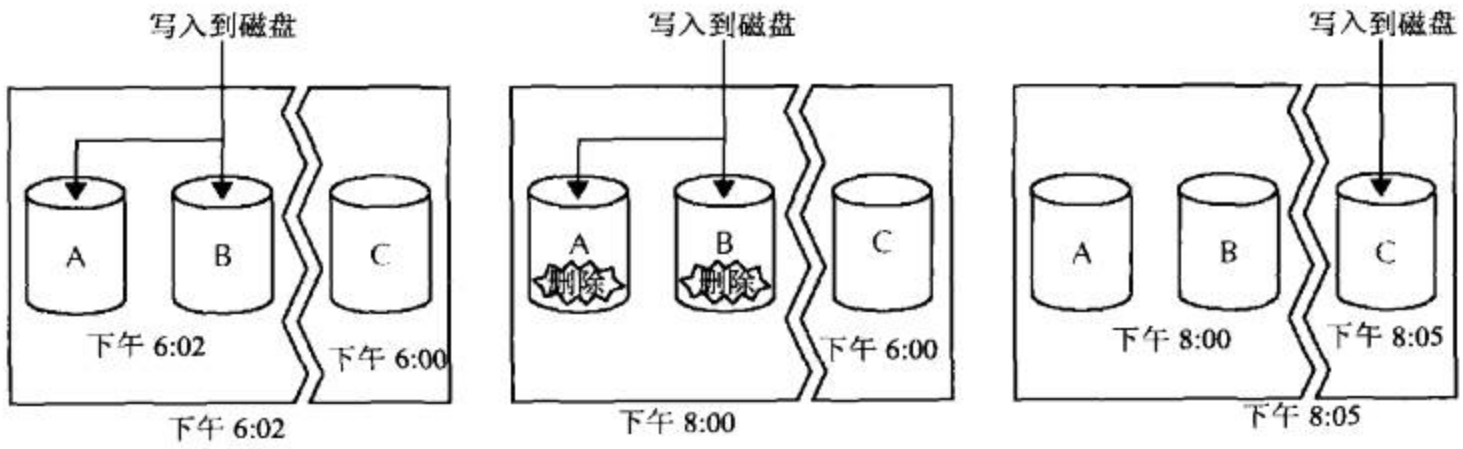


图 22-3 同步和拆分的实际操作过程

22.2 同步和拆分卷上的 Oracle 数据库

Oracle 软件文件可以驻留在同步和拆分卷上，因此可以避免在两个卷上出现逻辑损坏。从 Oracle 的观点来看，不需要任何额外的配置。另一方面，在将与 Oracle 数据库关联的文件放入同步和拆分卷中时，就会产生一些非常特别的警告和拒绝信息。这些警告和拒绝信息与如下情况相关：Oracle 文件总是保持打开状态，并且总是执行积极的写入操作(对于优秀的关系数据库，这一点尤其重要)。因此，如果积极地写入数据库，且该数据库正在两个驱动器上进行镜像，则突然断开镜像时就会导致数据库未知的结果。

每个供应商特有的解决方案都会稍微有所区别，但在某些方面却是相同的：对于正在进行积极写入操作的卷，必须关闭对该卷的写入，并继续允许写入到另一个卷。不论销售人员如何进行宣传，断开镜像的过程总是会持续一段时间。断开镜像非常类似于剥香蕉——从头部开始将香蕉皮与香蕉分离，直到剥到底部。假设 Oracle 数据文件是香蕉，数据文件的镜像副本是香蕉皮。如果要剥离镜像副本，则从数据文件的开始位置进行剥离，当到达数据文件的末尾时完成剥离操作。然而，非常可能发生的情况是：在镜像仍然处于剥离过程中时 Oracle 试图写入一个数据块。此时，在主卷上将不存在任何错误——即文件头知道 SCN 在文件前端，因而知道修改的具体位置；但是在拆分镜像上，数据文件头并不了解写入的那个数据块。因此，在镜像断开完成后，我们在拆分镜像卷上得到的只是一个不可恢复的模糊数据文件。图 22-4 实际地演示了这种情况。

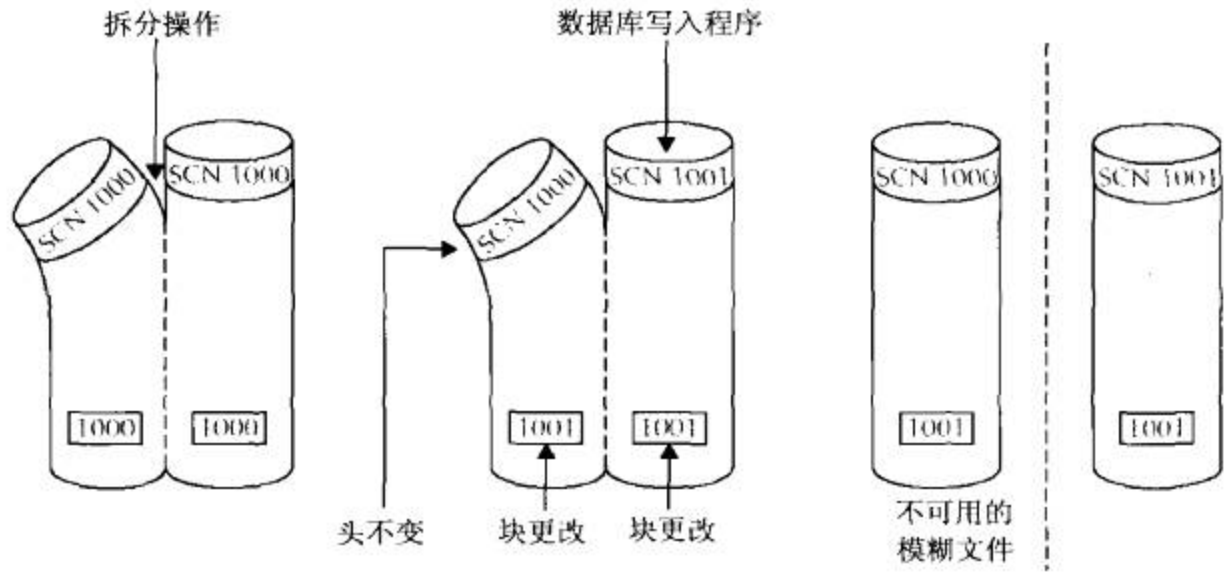


图 22-4　不可恢复的模糊数据文件

对此也不必惊慌，有一些方法可以确保拆分镜像是数据库的健康副本。首先只需要完成一些工作。并且在同步和拆分环境中的 Oracle 数据库文件配置取决于配置的文件类型：数据文件、控制文件、重做日志文件或归档日志。下面的小节将依次介绍这些文件类型。

22.2.1　数据文件

前面的小节介绍了在没有任何准备的情况下进行镜像拆分时 Oracle 数据文件上发生的情况：文件的拆分卷副本变成模糊的、不可用的状态。如果简单地获得联机数据文件的副本，而没有首先将其放入热备份模式，则会遇到相同的问题。因此，在断开镜像之前，必须将所有数据文件放入热备份模式。无论使用何种供应商产品，这都是一个必要的步骤。因为拆分操作一般只需要花费非常短的时间，而如果对相同的数据文件执行复制操作，热备份模式中的时间将会更短。而且在备份模式(并且产生较多的归档日志)下的 I/O 命中率也将会相对较小。

为了减少热备份模式为实现同步和拆分体系结构的操作引入的问题，Oracle 添加了相应的语法，使用一条命令就可以将整个数据库放入热备份模式：

```
alter database begin backup;
```

在此之前，我们必须将每个表空间放入热备份模式。如果某些方面阻止文件进入备份模式，则会在警报日志中生成一个警告信息，然后继续执行 begin backup 命令。

拆分完成后，使用如下命令将数据库从热备份模式中取出：

```
alter database end backup;
```

22.2.2　控制文件

拆分镜像操作完成后，控制文件的拆分镜像副本就会立刻处于不可用的状态。控制文件一般针对所有数据文件的当前状态保持更新。然而，根据拆分操作的总持续时间以

及进行拆分时数据库的全部活动，拆分卷中的控制文件可能不会反映数据文件状态的精确信息。

将数据文件放入热备份模式可解决大多数的问题。当数据库处于热备份模式时，控制文件就可知道需要恢复的起始位置，并且从该位置中可以进行切实可行的恢复。然而，控制文件仍然可能会无法确定它是否是活动数据库的当前控制文件。但这种情况很少见。

DBA 有时会坚持在拆分卷上保持当前控制文件可用，特别是在拆分卷用于报告目的时。然而，当开始将这个控制文件用于恢复操作时，我们必须使用 using backup controlfile 命令，从而使控制文件明白其中一些检查点和 SCN 信息未能反映实际的情况：

```
recover database using backup controlfile until cancel;
```

如果出于报告目的而在拆分镜像卷上加载 Oracle 数据库，则即使不应用任何归档日志，也可能需要使用上面的命令，而控制文件被标记为备份。我们将在 22.3 节中详细讨论这一点。

22.2.3 重做日志文件

联机重做日志的拆分镜像副本并没有什么作用。如果可能的话，最好不要将它们放入将要进行同步和拆分的卷。在联机重做日志中没有任何机制来解决拆分操作期间的文件写入。

22.2.4 归档日志

我们可以将归档日志放入同步和拆分卷中，这样做可得到磁盘上另一个位置中的已有归档日志的备份。当然，如果在拆分数据文件卷的同时拆分归档日志卷，则不会从拆分卷中获得恢复数据库所需的所有重做。我们建议将归档日志与数据文件和控制文件保存在一组不同的同步和拆分卷上。通过这种方式，我们可以拆分数据文件，将数据库从热备份模式中取出，强制进行日志切换，然后拆分归档日志卷。这样，具有归档日志的拆分镜像卷就包含启动数据库的拆分镜像副本所需的所有重做。

关于拆分镜像卷上的归档日志还需要注意一点：当数据库开始在磁盘上创建归档日志时，拆分操作可能在拆分镜像卷上留下未完成的归档日志。在任何恢复操作期间，这种归档日志将不可用。这就引发了一个人为管理备份和恢复操作的问题，在这种操作中并不清楚磁盘上的归档日志是否是已经完成，或者只是完成了一半。下面是 RMAN 中没有引发该问题的原因：当生成归档日志时，直到归档日志完成时才使用表明该归档日志存在的记录更新控制文件。因此，在拆分镜像的过程中，如果拆分卷上的归档日志只生成了一半，则该拆分卷上的控制文件将不包含该归档日志的任何记录。由于在 RMAN 操作期间将在控制文件中查询归档日志记录，而只完成一半的文件将不会存在于元数据中，所以根本找不到。对于 RMAN 来说，只完成一半的文件并不是真正存在的文件。

22.3　拆分镜像备份的优点

我们已经简要讨论了使用同步和拆分体系结构的主要优点：针对特定磁盘组上所有文件的即时回退恢复点。这种优点可以扩展到本书以外的范围(即 Oracle 数据库之外的应用)，包括卷上已有的所有文件的回退点。同步和拆分也有其他的主要优点，下面分别进行介绍。

22.3.1　快速时间点恢复

从数据库的观点来看，同步和拆分提供了只花费数分钟而非数小时的时间点恢复选项。我们只需要将主磁盘组改为拆分镜像，并且数据文件已经就绪。然后，应用归档日志，直到发生故障，接下来就可以打开数据库。

22.3.2　相对快速的备份

同步和拆分体系结构的另一个优点是相对快速的备份操作。当数据库处于热备份模式时，在拆分镜像中适当生成数据库文件的副本只需要花费很短的时间。完成该操作后，就可以非常快速地使用备份。当然，同步和拆分技术中没有任何魔法，仍然需要相同数量的 I/O。虽然备份操作看起来很快就可以完成，但实际情况是，备份占用了硬件级上的所有时间，因为在拆分操作之前同时写入这些文件。然而，很多应用都可以证明：硬件除了负责备份数据库之外还可以备份其他内容，将备份交给硬件体系结构负责是强有效的功能。

22.3.3　在另一个服务器上加载拆分镜像

除了简化还原和恢复操作的功能之外，当前市场上各种同步和拆分解决方案的许多实际功能来自于使用数据库的拆分副本可以完成的操作。底层的硬件可能是连接许多计算机的存储阵列，因此在理论上，该存储阵列上的任何卷与所有连接的计算机关联。

例如，获得一个数据库 PROD，该数据库驻留在卷 A 中的磁盘上，卷 B 对卷 A 进行镜像。卷 A 和卷 B 都连接到服务器 Dex，并且都位于存储阵列 Newton 上。在下午 2 点时，卷 B 与卷 A 进行拆分，并且取消与服务器 Dex 的关联。在此之后，立刻在不同的服务器 Proto 上加载卷 B，并且也连接到存储阵列 Newton。在 Proto 上加载卷 B 后，驻留在 Dex 上的数据库 PROD 的副本现在驻留在 Proto 上，并且具有几乎实时的数据量。可以恢复卷 B 上由服务器 Proto 加载的数据库副本，然后打开该副本进行测试、开发或报告。在下午 6 点，将卷 B 与卷 A 重新同步时，Proto 可以取消加载卷 B，然后由 Dex 重新加载卷 B。同步操作将重写下午 2 点的拆分操作之后发生的任何更改。

注意在打开处于不同节点上的数据库拆分镜像副本前，需要获得并使用新的备份控制文件。当卷 B 与卷 A 重新开始同步时，这个新的副本将会被 A 的正确的控制文件重写。

22.3.4 从拆分镜像中获得备份

在 RMAN 环境中，同步和拆分备份的另一个优点是能够在不同的服务器上加载拆分卷，并且从中将数据库备份到长期的存储介质——磁带上。这就允许我们将 RMAN 备份的内存、CPU 和 I/O 操作转交给完全不同的服务器，并且对产品数据库没有任何影响。

22.4 RMAN 与同步和拆分

RMAN 与同步和拆分实现具有一些不同的接触点：

- 如果使用 RMAN 进行恢复，必须使 RMAN 知道由拆分操作创建的数据文件副本。
- 可以使用 RMAN 从拆分镜像卷中获得备份，而不是从产品数据库中获得备份。

22.4.1 使用 RMAN 注册拆分镜像副本

如果只使用 RMAN，则会理解从 RMAN(而不是从 SQL*Plus 或其他位置)中执行所有恢复语句的优点。RMAN 恢复提供了对控制文件中的信息的访问，从而可以很快找到已有的备份，并且确保没有遗漏任何文件。控制文件也可以帮助进行恢复期间的归档日志管理。当同步和拆分系统准备好时，RMAN 并不知道任何相关的信息。拆分镜像卷的操作有效地提供了数据库中每个数据文件的完全数据文件副本，在还原/恢复操作期间可以使用该副本，但是 RMAN 并不知道这些副本的存在。

因此，必须使 RMAN 了解这些信息，其方法是使用 catalog 命令，使 RMAN 注册数据文件副本。可以对一个数据文件副本使用如下 catalog 命令：

```
catalog datafilecopy '/volumeA/oradata/system01.dbf';
```

或者，从 Oracle Database 10g 开始，可以按目录名对整个目录编写目录：

```
catalog start with '/volumeA/oradata';
```

通过使用 catalog 命令，我们就可以获得拆分镜像副本，并且将它们作为将来还原或恢复操作的一部分。

我们可能提出如下的问题：“当将整个卷作为主卷重新加载，并且在没有 RMAN 的帮助下启动和运行时，为什么需要让 RMAN 了解拆分镜像卷？”非常好的问题。但是可以设想一下，切换到文件的唯一副本是否更有意义？执行完全的数据库时间点恢复或许过于昂贵，但是我们仍然需要利用文件子集的拆分镜像副本。除此之外，RMAN 也极大地简化了任何操作的恢复阶段，因此使 RMAN 也了解归档日志的副本非常有意义。

22.4.2 从拆分镜像中获得 RMAN 备份

随着使用频率的不断增加，DBA 认识到需要额外的保护措施，即使用数据库的 RMAN 备份。根据定义，拆分镜像备份是短期存在的备份——当相应的卷与主数据库卷重新同步时，该副本就会丢失。但是，如果要从昨天晚上还原呢？或者从上个星期还原？

由此可见，仍然需要完整的介质备份。

当数据库的空闲副本对于拆分镜像来说并不重要时，DBA 就会产生如下的想法："我应该在不同的服务器上只加载拆分镜像驱动器，从拆分镜像中获得 RMAN 副本，并且直接放入磁带(或者是可以在主卷上加载的不同磁盘卷)"。非常好的想法！并且听起来也非常简单！但是，我们需要先解决一些难点，否则备份就有可能会突然消失！

下面是相关的问题：RMAN 访问控制文件以确定哪些内容需要备份，并在备份完成后，RMAN 使用备份的详细信息更新控制文件。如果连接到控制文件的拆分镜像副本，就使用备份的详细信息更新该副本。因此，在同步拆分卷和主卷时，控制文件中的内容将被主控制文件中的数据重写，并且备份数据就会永久地丢失。

解决方案是在拆分镜像上进行备份时使用恢复目录。这是合理的、逻辑的决策：在备份完成后，使用备份记录更新拆分卷控制文件，该控制文件然后与恢复目录同步。接下来，只需要将恢复目录与主卷同步，就可以使用该备份。

因此，假设从次卷中备份，我们将备份记录与恢复目录同步，然后将 RMAN 与主卷数据库和恢复目录进行连接。接下来执行重新同步。此时就可能发生一些预料之外的情况。有时，当尝试执行操作时会获得如下错误：

```
RMAN-20035: invalid high recid
```

而在其他时候，工作似乎正常进行，但是在拆分镜像数据库中获得的备份已经从恢复目录中消失。

现在，问题转变成 RMAN 如何处理在控制文件和恢复目录中构建的记录的内部机制。生成的每个记录获得一个在控制文件中生成的记录 ID(RECID)。在拆分镜像数据库中进行备份操作时，控制文件更新它的高 RECID 值，并且将该信息传递给恢复目录。但是，主数据库控制文件中的 RECID 不一定被更新。因此，当连接到恢复目录和主数据库时，如果恢复目录的高 RECID 大于控制文件中的 RECID，我们就会获得"invalid high recid"错误。如果恢复目录中的 RECID 小于主数据库控制文件中的 RECID，RMAN 就会初始化恢复目录的更新，从而有效地消除从上一次与主控制文件的同步操作以来的所有记录，也使得拆分卷中的备份记录消失！

解决这种问题的方案是设置拆分镜像中的控制文件为备份控制文件。如果 RMAN 检测到它正在从非当前的控制文件(备份或备用)中备份，它就不会增加恢复目录中的 RECID 值，这使得与主数据库中的当前控制文件重新同步后才可以使用备份记录。

如果没有从拆分镜像卷中获得备份，则不可以使用控制文件自动备份功能。因为使用中的控制文件是备份控制文件，因此禁止自动备份。

RMAN 实习：配置 RMAN 以从拆分镜像中备份

实习注意事项

该实习假设在拆分操作期间将所有表空间放入热备份模式中(必要条件)。拆分操作完成后，将拆分卷连接到安装 Oracle Database 10g 的新服务器，并且需要获得 RMAN 备份。因为 RMAN 在文件处于备份模式时将给出一个错误，因此需要手动结束每个文件的

备份，该实习中将进行该操作。最好为此编写一个脚本。另外，也假定要拆分归档日志目标，同时将其放入数据库副本以进行归档日志备份。

步骤 1 在复制服务器上加载数据库，并且为 RMAN 备份准备控制文件：

```
startup mount;
alter database end backup;
recover database using backup controlfile until cancel;
cancel
exit
```

步骤 2 将 RMAN 连接到复制实例(作为目标数据库)和恢复目录，并且运行数据文件备份：

```
rman target /
rman> connect catalog rman/password@rman_cat_db
rman> backup database plus archivelog not backed up two times;
```

步骤 3 将 RMAN 连接到产品数据库(作为目标数据库)和恢复目录，执行同步操作和归档日志清除操作，然后备份控制文件：

```
rman target /
rman> connect catalog rman/password@rman_cat_db
rman> delete archivelog completed before sysdate -7;
rman> backup controlfile;
rman> resync catalog;
```

22.4.3 廉价获得同步和拆分功能

如果有一种硬件解决方案能实现本章所描述的体系结构，那么它将具有相当大的优势。一般来说，相比于由软件完成的操作，完全在硬件级上完成的相同操作具有更好的性能。出于同样的原因，硬件解决方案总是比软件解决方案需要更多的成本。同步和拆分解决方案同样如此——在存储阵列上完成的工作越多，则工作完成的速度就越快——需要的成本就更多。

从 Oracle Database 10*g* Release 2 开始，Oracle 提供了完全的解决方案，该解决方案提供了同步和拆分功能，而不需要为任何第三方硬件或软件解决方案付费，我们只需要有 Oracle Database 10*g* 企业版、两个服务器(有相同 OS)以及一个存储阵列就行了。

22.4.4 将备用数据库、闪回数据库以及增量应用于同步和拆分

为了只使用 Oracle 软件实现同步和拆分解决方案，我们需要使用 RDBMS 中的不同功能集：备用数据库、闪回数据库以及 RMAN 增量备份和增量应用。在前面的章节中已经在一定程度上讨论了所有这些功能。

下面是工作原理。首先创建产品数据库的备用数据库(参见第 20 章中的实习)。一旦备用数据库可作为灾难恢复解决方案，就需要在产品数据库和备用数据库上实现闪回数

据库：

```
alter database flashback on;
```

启用闪回数据库后，我们就可以在主服务器上设置还原点：

```
create restore point chapter_20;
alter system switch logfile;
```

通过还原点将更改应用于备用数据库。此时，备用数据库处于打开状态(具有复位日志)以进行测试和报告：

```
alter database activate standby database;
```

为了让备用数据库与主数据库重新同步，我们需要获得增量备份，同时使用 from scn 关键字来指定还原点的 SCN。一旦完成该备份，就将其移动到备用数据库站点。

```
backup database incremental from scn = 120000;
```

在备用数据库中，关闭该数据库，然后再次重新加载数据库。执行闪回数据库，闪回到在打开备用数据库之前指定的还原点。

```
flashback database to restore point chapter_20;
```

一旦闪回操作完成，将产品数据库的增量备份应用于备用数据库，恢复到备份的适当位置：

```
recover database until scn=1521321;
```

然后，备用数据库可以重新处于受管理的备用模式，并且与产品数据库同步。或者，它可以再次处于打开状态以进行报告，并且现在具有通过增量备份导入的所有最新的数据。图 22-5 显示了该进程的工作原理。

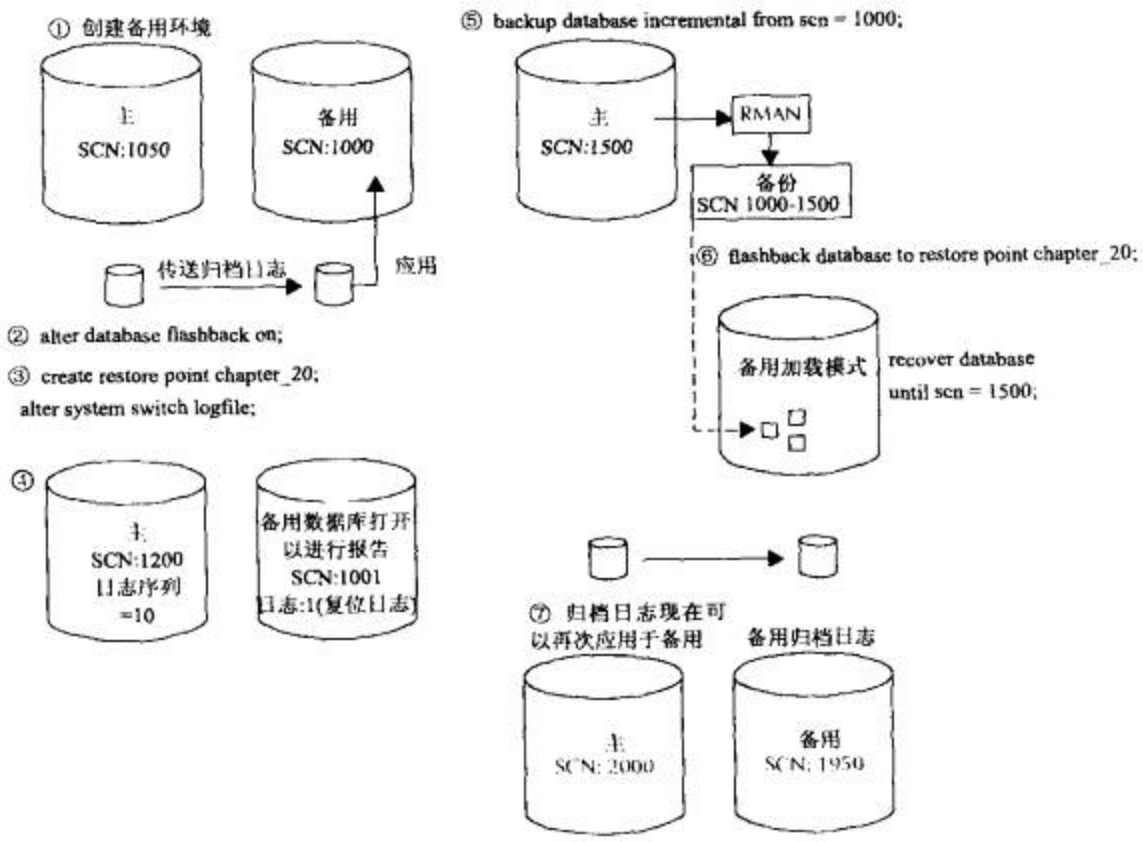


图 22-5　结合备用数据库和闪回数据库使用同步和拆分

22.4.5 Oracle 同步和拆分解决方案的优点

使用 Oracle 同步和拆分解决方案的唯一优点在于只需要较少的成本。虽然使用备用数据库/闪回数据库/增量应用解决方案会降低性能，但这些性能降低可能并不像我们想象的那么严重，它完全取决于是否已经将闪回日志用于这些解决方案提供的固有功能。如果已经这样做，则有两个数据库更改的日志：闪回日志和重做日志。在文件系统级上执行更多的日志记录只会添加额外的、冗余的日志，可以去除这些日志。

此外，现在我们有一个可以用于灾难恢复的备用数据库。虽然灾难恢复也是硬件同步和拆分模型中的固有功能，但是有一个可以自由支配的备用数据库会极大地自动化和简化在实际灾难恢复期间的手动操作。

最后，决定使用完全的 Oracle 解决方案或硬件解决方案也需要考虑其他因素。Oracle 数据库之外的其他方面是否也需要同步和拆分体系结构？是否具有其他企业版本数据库的授权？是否有使用某种解决方案的经验？显然，我们需要说明这些问题。除此之外，我们也需要彻底地测试解决方案。关于 Oracle 解决方案的好消息是，我们已经具有了立刻测试该解决方案的所有必要条件。

22.5 应用于 Windows 的 Oracle 集成卷影复制服务

在集成的 Oracle 11g 中可以看到一种硬件/操作系统集成的同步和拆分类型的方向，一个有意思的实例是 Windows 平台上的卷影复制服务功能(VSS，Volume Shadow Copy Service)。VSS 涉及背景日志功能，类似其他供应商的镜像功能，背景日志可以拆分为不同的卷，并保存到存储阵列的不同位置。VSS 作为 Windows 操作系统的一个组件提供了协调存储写入(Oracle 数据库)和存储提供(存储阵列技术)之间活动的能力。该功能可协调基于组件的卷影复制，意味着不必仅仅考虑卷集。VSS 可以通知卷上的组件并执行相应的操作。

Oracle 为 VSS 创建了一个称为 Oracle VSS 写入器的插件，该插件是一个独立于 Oracle 数据库服务的 Windows 服务。Oracle VSS 写入器负责协调获取数据库 VSS 副本的特定操作。

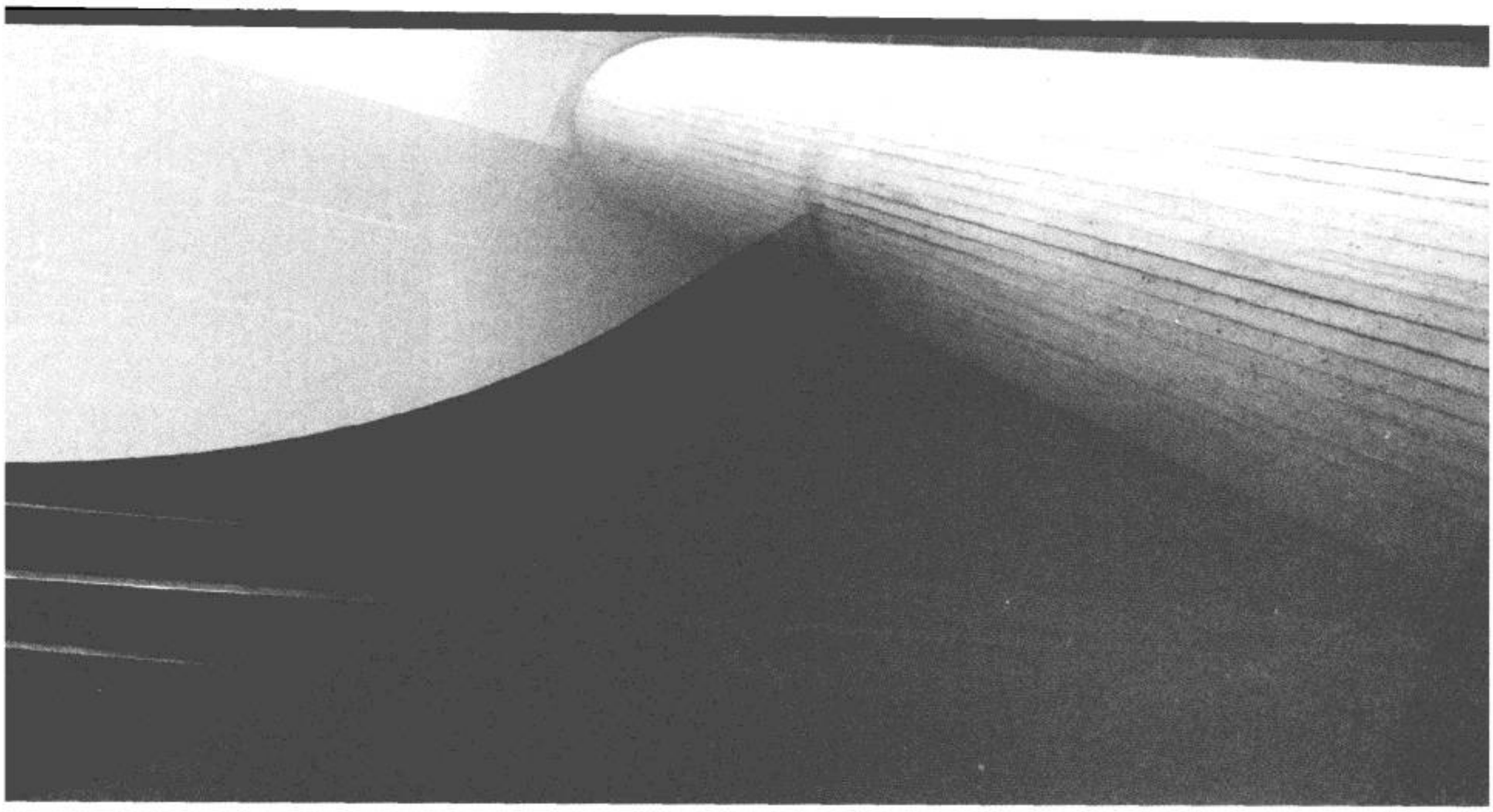
Oracle VSS 写入器可以创建组件级别的备份(如文件对文件，如数据文件与控制文件)，也可以创建全卷备份。在创建数据文件的组件级别的备份时，VSS 写入器将跟踪不同的现存机制创建的重做，在还原期间，将重做自动地应用到备份的组件。

VSS 建立全卷备份时，数据库的数据块在写入过程中仍然会被 VSS 写入器获取，因此可能会是不确定的。因此在数据文件备份期间，Oracle VSS 写入器将以与前面讨论的类似的方法执行：将数据文件置入热点备份模式以进行数据文件的持久备份，以便归档日志拥有更改的完全副本以重写所有模糊的块。

我们将会看到，集成级别的差别是—当同步和拆分技术提供了更好的接口点，正如 Microsoft 所做的那样，允许 Oracle 提供更好的任务自动化技术，如若不然，系统管理员或者数据库管理员不得不分别创建脚本执行操作。

22.6　小结

本章介绍了硬件同步和拆分体系结构影响备份和恢复解决方案的方式，并且讨论了如何使用 Oracle 数据库实现同步和拆分，以及如何从数据库的拆分镜像副本中获得 RMAN 备份。最后，本章讨论了如何使用已有的 Oracle RDBMS 实现基于软件的同步和拆分环境。



第 23 章

RMAN 案例研究

本书已经介绍了许多不同的主题，但是您要面对的恢复问题是层出不穷的。本章将通过一些不同的案例研究来帮助您复习备份与恢复的有关内容(查看是否可以在使用之前指出解决方案)。如果遇到类似的情况，这些案例研究会帮助您在恢复数据库时消除错误。您也可以使用这些案例研究来练习执行恢复操作，这样可以熟悉和精通 RMAN 的备份与恢复。

但在进入案例研究前，我们先讲述一下在数据库出现故障时以及在执行恢复操作前应首先考虑哪些问题。

23.1 恢复操作前的准备工作

试想一下灾难真的来时的情形，通常此时每个人都会急于恢复数据库。用户的不断问询、管理层的恐慌以及老板的督促会使您非常紧张，所有人都迫切希望尽快恢复数据库。当出现需要执行恢复操作的情况时，您一定不要着急，而应当先考虑下面这些问题：

(1) 故障的本质何在？

(2) 可以使用哪些恢复选项？

(3) 需要 Oracle 的支持吗？

(4) 在恢复操作期间，有能够帮助自己的人吗？

下面将详细讨论这些问题。

23.1.1 故障的本质

某次某个站点的一台服务器出现故障，在服务器重新运行后，数据库却无法启动。这个站点的 DBA 花费了近 8 个小时来尝试启动这台服务器上的 25 个数据库。大多数数据库不能启动。DBA 恢复了几个看上去丢失的数据库，但是这些数据库仍然不能打开。DBA 给 Oracle 打电话，但是 Oracle 也不能确认所出现的问题。最后，我接到他们的电话。

到达办公室后的 20 分钟内，基于两个原因，我发现了问题的实质。第一个原因是我从一个新的角度出发来考虑问题(不同于 8 小时奋战后异常疲惫的 DBA 的思维)；第二个原因是我努力发现故障的实质，而不是假定故障是由于数据库损坏。

问题的根本原因在于一组 Oracle 库出现损坏。一旦恢复了这些库，就能够无故障地启动所有数据库。这个例子的教训是：数据库发生灾难或不能打开时，不要假定故障是由数据文件损坏或磁盘驱动器损坏所引起的。我们应当通过研究分析来发现问题的实质，认真的分析在最开始时可能会花费一些时间，但是通常会取得巨大的回报。

23.1.2 可用的恢复选项

恢复情况可以提供许多不同的解决方案。过去，我是一名咨询人员，有一位用户的磁盘控制器驱动器在周末出现故障，并导致机器上的文件系统丢失(包括属于一个 ARCHIVELOG 模式的 Oracle 数据库的文件)。该用户所在站点的 DBA 恢复了整个数据库(大约 150GB)，据我回忆，这个恢复操作花费了数个小时。

接下来的星期一，我和这位 DBA 讨论了他可以选择的恢复方法。损坏的文件系统实际上只影响 5 个数据库数据文件(我们不必关心包含 Web 服务器文件的其他文件系统)。受影响的数据库数据文件的全部大小只有 8GB 或 10GB。这位 DBA 对花费了几个小时来恢复整个数据库，对此，他也感到十分不满。我询问他为什么不只还原这 5 个数据文件而是还原整个数据库时，这位 DBA 回答说自己没想过这种方法。

这个例子的教训是：考虑不同的恢复选项是非常重要的。选用的恢复类型可能会使恢复数据库所花费的时间大不相同。这个例子的另一个教训是：要尽可能地精通 RMAN 的备份与恢复的各种技术。我认为，这位 DBA 没有考虑数据文件恢复选项的部分原因是

他还不熟悉 RMAN 的恢复操作。在紧急的情况下，人们大多不会考虑使用自己不熟悉的恢复选项。因此，我们强烈建议您反复练习 RMAN 的备份与恢复操作，直至熟练为止。

23.1.3　可能需要的 Oracle 支持

您可能很精通 RMAN 备份与恢复，不过即便如此，偶尔也会需要得到一些帮助，这就是需要 Oracle 支持的原因。即使自我感觉掌握了备份与恢复的有关知识，我们仍然需要知道是否需要 Oracle 的支持。通常，如果故障是不确定的，即使我认为自己能够解决问题，我也会通过打开该问题的服务请求获得支持。这样，如果需要帮助，我就可以先为 Oracle 提供信息(至少要提供一些初始信息)。如果您付款获得了 Oracle 支持，就应当立刻使用这个支持。

23.1.4　在恢复操作期间寻求帮助

在紧急的情况下，您最好首先找到一个能够共同承担压力的人。不知什么原因，有人和我说话时，我会觉得轻松一点。进一步说，在处理关键问题时一旦出错，代价会很大。恢复数据库时寻求一个有经验的人的帮助，这是非常好的一个做法。

23.2　恢复案例研究

本章主要讨论的是恢复案例研究。本节将按顺序介绍下面这些案例：

(1) 在 NOARCHIVELOG 模式中恢复完全丢失的数据库(使用恢复目录)
(2) 在 NOARCHIVELOG 模式中恢复完全丢失的数据库(不使用恢复目录)
(3) 在 ARCHIVELOG 模式中恢复完全丢失的数据库(使用恢复目录)
(4) 在 ARCHIVELOG 模式中恢复完全丢失的数据库(不使用恢复目录)
(5) 恢复丢失的 SYSTEM 表空间
(6) 联机恢复丢失的数据文件或表空间
(7) 恢复丢失的未归档的联机重做日志
(8) 使用 resetlogs 命令的恢复操作
(9) 手动完成失败的复制操作
(10) 使用 RMAN 复制创建目标数据库的历史子集
(11) 在 ARCHIVELOG 模式中使用闪回恢复区中的映像副本恢复丢失的数据文件
(12) 执行在闪回恢复区之外运行产品数据文件的恢复
(13) 使用闪回数据库和介质恢复来指出使用 resetlogs 命令打开数据库的确切时间

在这些案例中，我们提供了下面这些信息：

- 概况：概述所提供的环境。
- 问题：定义需要解决的问题。
- 解决方案：概述要使用的解决方案，包括解决问题的 RMAN 输出。

案例 1：使用恢复目录恢复完全丢失的数据库(NOARCHIVELOG 模式)

概况

Thom 是 Unfortunate 公司的新 DBA。接手这个工作时，他发现数据库根本没有被备份，并且这些数据库都被置于 NOARCHIVELOG 模式。由于经理不打算购买用于存储归档重做日志的磁盘，所以 Thom 只能执行脱机备份操作，这就是他第一天晚上的工作。Thom 还启用了控制文件的自动备份，并且使数据库使用服务器参数文件。最后，Thom 在不同数据库服务器上的不同数据库中创建了一个恢复目录模式。

问题

Thom 做完最初工作后的某一天，Unfortunate 公司的廉价设备出现了问题。由于控制器出现故障，Thom 的数据库丢失了。

Thom 的脱机数据库备份策略包括在本地磁带驱动器上的磁带备份。一旦解决了硬件问题，系统管理员很快地重新构建了丢失的文件系统，同时 Thom 迅速地安装了 Oracle 软件。此时，Thom 需要尽快备份并运行数据库。

解决方案

在这种情况下，Thom 可以选用的唯一恢复选项是从最后一个脱机备份中还原数据库(如第 12 章所述)。此时，恢复目录数据库并未丢失(该数据库位于另一个服务器上)，并且文件系统也是正确的，所以 Thom 只需要恢复数据库。首先，他需要恢复数据库的服务器参数文件和控制文件。然后，还需要将数据库数据文件恢复到文件系统中。

解决方案的实现

基于前面的这些考虑，Thom 设计和实现了下面的恢复计划：

(1) 还原服务器参数文件的副本。虽然很多情况下没有服务器参数文件也能以未加载模式启动 Oracle 实例，但为了正确恢复数据库，Thom 必须从备份中还原正确的服务器参数文件。由于还没有控制文件，所以不能实现对信道的永久配置。在这个案例中，Thom 配置为在默认的磁盘位置自动备份控制文件。这样，只要 Thom 还原了 Oracle 软件备份，就可以将备份片还原到控制文件的自动备份中。这使还原服务器参数文件的操作非常简单：

```
rman target sys/password catalog rcat_user/rcat_password@catalogdb
startup force nomount;
restore spfile from autobackup;
shutdown immediate;
startup nomount;
```

注意：

如果没有使用 FRA，则需要在执行服务器参数文件和控制文件的还原时，设置数据库的 DBID。

(2) 还原控制文件的副本。使用与步骤 1 相同的 RMAN 会话就可以完成该操作。完成还原操作后，Thom 使用这个还原的控制文件加载数据库。

```
restore controlfile from autobackup;
alter database mount;
```

(3) 配置永久的信道参数。由于还原了控制文件，Thom 可以更新用于信道分配的永久参数包括备份集所在磁带设备的名称，这样可以继续还原磁带备份和恢复数据库，如下所示：

```
configure default device type to sbt;
  configure channel 1 device type sbt
  parms = "env=(nb_ora_serv=mgtserv, nb_ora_client=cervantes)";
```

(4) 执行下面的还原和恢复操作：

```
restore database;
recover database noredo;
alter database open resetlogs;
```

注意：
Thom 使用 alter database open resetlogs 命令，他本来也可以使用 SQL 命令 sql"alter database open resetlogs"。然而，使用 RMAN 的 alter 命令的一个优点是同时复位了恢复目录和数据库。而使用 SQL 版本的命令时，只会复位数据库。

案例 2：不使用恢复目录恢复完全丢失的数据库(NOARCHIVELOG 模式)

概况

Charles 是一个开发 OLTP 系统的 DBA。由于这是一个开发系统，所以确定要执行 RMAN 脱机备份操作，并且将数据库置于 NOARCHIVELOG 模式中。Charles 不打算在执行备份操作时使用恢复目录。此外，Charles 配置 RMAN 为默认地将控制文件备份到磁盘上(而不是磁带上)。

问题

开发人员 Sevi 负责开发设计截断数据库中特定表的一段 PL/SQL 代码。但是，由于逻辑错误，这段代码截断了模式中的所有表，删除了所有测试数据。

解决方案

如果存在数据库的逻辑备份，就可以使用该备份。但是，这个案例不存在数据库的逻辑备份，因此 Charles(DBA)需要执行 RMAN 恢复操作。由于数据库被置于 NOARCHIVELOG 模式中，所以 Charles 只能选用唯一的恢复选项，即还原最后一个脱机备份(参阅第 12 章)。由于用于恢复的所有部分(RMAN 磁盘备份、Oracle 软件以及文件系统)都已到位，所以 Charles 只需要启动 RMAN 和恢复数据库。

解决方案的实现

基于前面的这些考虑，Charles 设计和实现了下面的恢复计划：

(1) 还原控制文件。恢复冷备份时，最好要还原与这个备份关联的控制文件(这样可以防止不确定的事情发生)。在这个案例中，Charles 会使用最新的控制文件备份(由于他在其他时间未备份控制文件)。由于 Charles 使用默认的位置来创建控制文件备份，所以不必分配任何信道。由于不使用 Oracle 闪回恢复区和恢复目录，因此在还原控制文件前 Charles 需要设置系统的 DBID。如果他使用恢复目录或 FRA，则不需要设置系统的 DBID。一旦还原了控制文件，就可以加载这个数据库。

```
rman target sys/password
    startup nomount
    set dbid=2540040039;
    restore controlfile from autobackup;
    sql 'alter database mount';
```

注意：
如果使用 FRA，则不需要设置系统的 DBID。

(2) Charles 还原的控制文件已配置正确的默认永久参数，所以只需要执行下面的还原和恢复操作：

```
restore database;
recover database noredo;
sql "alter database open resetlogs";
```

案例 3：不使用恢复目录恢复完全丢失的数据库(ARCHIVELOG 模式)

概况

还是案例 1 的那个公司。Thom 的公司最后决定将数据库置于 ARCHIVELOG 模式中，Thom 的老板认为这是一个好方法。然而，对于 Thom 来说，由于预算的限制，他不得不占用恢复目录的空间来存储归档的重做日志，因此 Thom 不能使用恢复目录。

问题

除了 Thom 要面对的问题之外，Unfortunate 公司也遇到了麻烦。公司的服务器位于地下室内，由于附近的水管破裂，整个地下室被淹，数据库所驻留的服务器被完全破坏。

Thom 改进了备份策略，该策略现在包括在离站的介质管理服务器上执行磁带备份操作，此外，Thom 在磁带上(而不是在磁盘上)自动备份控制文件/服务器参数文件。Thom 抢修了一台较小的服务器，这个服务器的系统中安装了 Oracle 软件，接下来他需要尽快启动运行数据库。

解决方案

同样，Thom 丢失了用于数据库的当前控制文件和联机重做日志，因此需要使用第 14

章所介绍的时间点恢复技术。由于 Thom 启用了控制文件的自动备份，所以可以使用这些自动备份来启动恢复操作。此外，由于要在新的服务器上还原文件，所以 Thom 还要了解可能带来的问题，这些问题涉及介质管理、文件系统布局以及内存使用问题。

介质管理问题

由于要在新的服务器上还原文件，所以 Thom 必须首先确保正确创建了在应急服务器上使用的 MML 文件，也就是说在使用 RMAN 执行备份与恢复操作前，应当先安装 Media Management Client 软件和 Oracle Plug-In 插件。Thom 使用 sbttest 实用程序来检查以确保介质管理器可访问(这是一个不错的方法)。

接下来，Thom 需要配置磁带信道以指定遭破坏服务器的客户端名。如第 4 章所述，Thom 必须指定生成备份的客户端名。此外，他还必须确保介质管理服务器配置为允许将不同客户端上的备份还原到应急服务器上。

文件系统布局问题

Thom 的新系统具有与原始服务器不同的文件系统结构。原来的产品数据库将所有文件手动分布在 6 个加载点中：/u02、/u03、/u04、/u05、/u06 和/u07。新的服务器只具有两个加载点：/u02 和/u03。幸运的是，Thom 在本企业中使用了目录结构标准，所有加载点上的数据目录都是/oradata/prod/。此外，他总是按照标准将 ORACLE_HOME 主目录放在每台服务器中相同的加载点和目录结构上。

内存使用问题

应急服务器的物理内存比丢失的产品服务器的物理内存少，也就是说 Thom 不得不重新考虑内存使用问题，以保证至少能够启动和运行数据库。

解决方案的实现

基于前面的这些考虑，Thom 设计和实现了下面的恢复计划：

(1) 确定目标数据库的 DBID。通过查找控制文件自动备份的文件句柄可以完成这个操作。Thom 需要能够查看介质管理目录，甚至还可以更简单地在日志中(如笔记本、PDA 等)存储所有数据库的 DBID。无论使用哪种工具，都需要确保能够在应急情况下访问这种工具。

(2) 还原服务器参数文件的副本。Thom 会使用一个虚拟的服务器参数文件打开实例，然后从备份中还原正确的服务器参数文件。因为 Thom 修改为在磁带上默认自动备份控制文件/服务器参数文件。并且，由于还没有控制文件，所以他必须为这个还原操作手动地配置信道，而不能永久地配置信道。如下所示，Thom 必须将 channel allocation 命令加入到 run 代码块中，然后通过执行 startup 命令使用正确的服务器参数文件启动数据库：

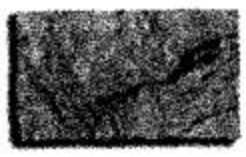
```
rman target /
 set dbid=204062491;
startup force nomount;
run {
allocate channel tape_1 type sbt
parms='env=(nb_ora_serv=rmsrv, nb_ora_client=cervantes)';
```

```
restore spfile from autobackup;}
shutdown immediate;
startup nomount;
```

注意：
如果使用 FRA，则需要设置数据库的 DBID。

(3) 修改服务器参数文件。Thom 必须根据新服务器的配置来修改服务器参数文件，也就是说要修改内存分配参数和设置文件名转换参数。必须从 SQL*Plus 连接最新启动的实例，并且做一些必要的修改。

```
alter system set control_files= '/u02/oradata/prod/control01.dbf',
'/u03/oradata/prod/control02.dbf' scope=spfile;
alter system set db_file_name_convert= ('/u04' , '/u02' ,
'/u05' , '/u02' ,
'u06' , ' u03' ,
'u07' , 'u03') scope=spfile;
alter system set log_file_name_convert= ('/u04' , '/u02' ,
'/u05' , '/u02' ,
'u06' , ' u03' ,
'u07' , 'u03') scope=spfile;
alter system set log_archive_dest_1=
'location=/u02/oradata/prod/arch' scope=spfile;
alter system set db_cache_size=300m scope=spfile;
alter system set shared_pool_size=200m scope=spfile;
shutdown immediate;
startup nomount;
```



注意：
在这里也可以选择使用 set newname 选项。

(4) 还原控制文件的副本。使用与前面相同的 RMAN 会话就可以简单地完成该操作(已经设置了 DBID)，然后再使用这个还原的控制文件加载数据库。

```
run {
allocate channel tape_1 type sbt
parms='env=(nb_ora_serv=rmsrv, nb_ora_client=Cervantes)';
restore controlfile from autobackup; }
alter database mount;
```

(5) 配置不变的信道参数。由于还原了控制文件，Thom 可以更新用于信道分配的持久参数，将丢失的服务器作为介质管理客户端的名称包括进来，这有两个目的：允许 RMAN 访问从丢失的服务器生成的备份；从新服务器生成备份时，RMAN 会将这个客户端名传递给介质管理服务器。这样，以后重新构建丢失的服务器时，在新构造的产品服务器上可以访问这个临时系统生成的所有备份。

```
configure default device type to sbt;
configure device type sbt parallelism 2;
configure auxiliary channel 1 device type sbt parms
= "env=(nb_ora_serv=mgtserv, nb_ora_client=cervantes)";
configure auxiliary channel 2 device type sbt parms
= "env=(nb_ora_serv=mgtserv, nb_ora_cient=cervantes)";
```

(6) 确定副本中的最后一个归档日志。由于丢失了整个服务器，Thom 也丢失了 RMAN 未备份的所有归档日志，因此他必须查询 RMAN，以确定已有备份中的最后一个归档日志，如下所示：

```
List backup of archivelog from time = 'sysdate-7';
```

(7) 使用现有的最后一个日志序列号，Thom 就可以执行下面的还原与恢复操作并打开这个数据库：

```
restore database;
recover database until sequence=<number>;
alter database open resetlogs;
```

案例 4：使用恢复目录恢复完全丢失的数据库(ARCHIVELOG 模式)

概况

Thom 的工作非常出色，所以公司奖励他和他的妻子去夏威夷旅游两个星期，在这期间，Charles 接手了他的工作。在 Thom 离开前，公司添置了磁盘存储设备，并决定使用 RMAN 的恢复目录。

然而，Charles 也遇到了麻烦。一只老鼠咬坏了通往计算机室的电力线，结果电线短路引起计算机室内的火灾，造成数据库所驻留的服务器和磁盘完全损坏。

问题

Charles 查看了 Thom 的备份策略。同样地，他抢修了一台幸存的较小服务器，这个服务器的系统中安装了 Oracle 软件，接下来他需要尽快启动并运行数据库。幸运的是，恢复目录服务器没有受到任何损伤，所以 Charles 可以在恢复操作期间使用恢复目录。

解决方案

同样，Charles 丢失了数据库的当前控制文件和联机重做日志，因此需要使用第 14 章所介绍的时间点恢复技术。由于备份策略启用了控制文件的自动备份，所以可以使用这些自动备份来启动恢复操作。此外，由于要在新的服务器上还原文件，所以 Charles 还要了解可能带来的问题，这些问题涉及介质管理、文件系统布局以及内存使用。

介质管理问题

由于要在新的服务器上还原文件，所以 Charles 必须首先确保正确创建了在应急服务器上使用的 MML 文件，也就是说在使用 RMAN 执行备份与恢复操作前，应当先安装 Media Management Client 软件和 Oracle Plug-In 插件。Charles 会使用 sbttest 实用程序来检查以确保介质管理器可访问。

接下来，Charles 需要配置磁带信道以指定遭破坏服务器的客户端名。如第 4 章所述，他必须指定生成备份的客户端名。此外，Charles 还必须确保介质管理服务器配置为允许将不同客户端上的备份还原到应急服务器上。

文件系统布局问题

Charles 的新系统具有与原始服务器不同的文件系统结构。以前的产品数据库将所有文件手动分布在 6 个加载点中：/u02、/u03、/u04、/u05、/u06 和/u07。新的服务器只具有两个加载点：/u02 和/u03。幸运的是，Charles 在本企业中使用了目录结构标准，所有加载点上的数据目录都是/oradata/prod/。此外，他总是按照标准将 ORACLE_HOME 主目录放在每台服务器中相同的加载点和目录结构上。

内存使用问题

应急服务器的物理内存比丢失的产品服务器的物理内存少，也就是说 Charles 不得不重新考虑内存使用问题，以保证至少能够启动和运行数据库。

解决方案的实现

基于前面的这些考虑，Charles 设计和实现了下面的恢复计划：

(1) 还原服务器参数文件的副本。首先，由于 Oracle 的支持，Charles 不用参数文件也不用加载数据库实例，然后再从备份中还原正确的服务器参数文件。由于还没有控制文件，所以 Charles 不能永久地配置信道。相反，他必须将 channel allocation 命令加入到 run 代码块中，然后通过执行 startup 命令使用正确的服务器参数文件启动数据库。由于使用了恢复目录，Charles 不必像上一个案例那样先设置目标数据库的 DBID。

```
rman target / catalog rcat_user/rcat_password@catalog
    startup force nomount;
    run {
    allocate channel tape_1 type sbt
    parms='env=(nb_ora_serv=rmsrv, nb_ora_client=cervantes)';
    restore spfile from autobackup;}
    shutdown immediate;
    startup nomount;
```

(2) 修改服务器参数文件。Charles 必须根据新服务器的配置来修改服务器参数文件，也就是说要修改内存分配参数和设置文件名转换参数。从 SQL*Plus 连接最新启动的实例，并且做一些必要的修改。

```
alter system set control_files= '/u02/oradata/prod/control01.dbf',
    '/u03/oradata/prod/control02.dbf' scope=spfile;
    alter system set db_file_name_convert= "('/u04' , '/u02' ,
    '/u05' , '/u02' ,
    '/u06' , '/u03' ,
    '/u07' , '/u03')" scope=spfile;
    alter system set log_file_name_convert= "('/u04' , '/u02' ,
    '/u05' , '/u02' ,
    '/u06' , '/u03' ,
```

```
'/u07' , '/u03')" scope=spfile;
alter system set log_archive_dest_1=
'location=/u02/oradata/prod/arch' scope=spfile;
alter system set db_cache_size=300m scope=spfile;
alter system set shared_pool_size=200m scope=spfile;
shutdown immediate;
startup nomount;
```

(3) 还原控制文件的副本。使用与前面相同的 RMAN 会话就可以简单地完成该操作(已经设置了 DBID)，然后再使用这个还原的控制文件加载数据库。

```
run {
allocate channel tape_1 type sbt
parms='env=(nb_ora_serv=rmsrv, nb_ora_client=Cervantes)';
restore controlfile from autobackup; }
sql 'alter database mount';
```

(4) 配置永久的信道参数。由于还原了控制文件，Charles 可以更新用于信道分配的永久参数，并将丢失的服务器作为介质管理客户端的名称包括进来，这有两个作用：允许 RMAN 访问从丢失的服务器生成的备份；从新服务器生成备份时，RMAN 会将这个客户端名传递给介质管理服务器。这样，以后重新构建丢失的服务器时，在新构造的产品服务器上可以访问这个临时系统生成的所有备份。

```
configure default device type to sbt;
configure device type sbt parallelism 2;
configure auxiliary channel 1 device type sbt parms
= "env=(nb_ora_serv=mgtserv, nb_ora_client=cervantes)";
configure auxiliary channel 2 device type sbt parms
= "env=(nb_ora_serv=mgtserv, nb_ora_cient=cervantes)";
```

(5) 确定副本中的最后一个归档日志。由于丢失了整个服务器，Charles 也丢失了 RMAN 还未备份的所有归档日志，因此他必须查询 RMAN，以确定已有备份中的最后一个归档日志，如下所示：

```
list backup of archivelog from time = 'sysdate-7';
```

(6) 使用现有的最后一个日志序列号，Charles 就可以执行下面的还原与恢复操作并打开这个数据库：

```
restore database;
recover database until sequence=<number>;
sql "alter database open resetlogs";
```

案例 5：恢复丢失的 SYSTEM 表空间

概况

Nancy 是一位令人敬畏的 DBA，她负责一个大型的数据库。为了使 Unix 的系统管理员能够对文件系统进行维护工作，Nancy 关闭了数据库。

问题

遗憾的是，在维护操作期间，公司的系统管理员删除了数据库所在的一个文件系统，最后管理员恢复了这个文件系统，但是数据库的所有文件已经丢失，所以 Nancy 必须恢复这些文件。Nancy 丢失了下列表空间中的所有数据文件：USERS、SYSTEM 和 INDEX 表空间。

解决方案

幸运的是，系统并未全部丢失，联机重做日志和控制文件保存完好。由于 Nancy 必须恢复 SYSTEM 表空间，所以她必须在数据库关闭时执行恢复操作。就其他方面而言，这个恢复操作还是非常简单的。

解决方案的实现

基于前面的这些考虑，Nancy 设计和实现了恢复计划，该计划只需要还原数据库，如下所示：

```
rman target / catalog rcat_user/rcat_password@catalog
startup force mount;
restore tablespace users, system, index;
recover tablespace users, system, index;
alter database open;
```

案例 6：联机恢复丢失的数据文件或表空间

概况

Yang 正在运行数据库时，由于电源不稳定导致介质出现故障。

问题

遗憾的是，Yang 丢失了一个文件系统。这个文件系统含有下面的这些数据文件：

- WORKING_DATA 表空间的所有数据文件
- HISTORICAL_DATA 表空间的一个数据文件

数据库中的其他一些表空间与要恢复的表空间没有关系，所以 Yang 需要在数据库打开和运行时执行这个恢复操作。

解决方案

Yang 要通过 RMAN 还原整个 WORKING_DATA 表空间和 HISTORICAL_DATA 表空间中丢失的单个数据文件。首先，他要使这些表空间和数据文件脱机，以便别人继续执行其他操作。

解决方案的实现

基于前面的这些考虑，Yang 设计和实现了下面的恢复计划：

(1) 如下所示，使 WORKING_DATA 表空间脱机：

```
sql "alter tablespace working_data offline";
```

(2) 如下所示，使 HISTORICAL_DATA 表空间中需要恢复的数据文件脱机(Yang 已经通过查询 V$DATAFILE 视图确定其为数据文件 13)：

```
sql "alter database datafile 13 offline";
```

(3) 如下所示，使用 RMAN 还原与恢复这个表空间和数据文件，然后使它们联机：

```
restore tablespace working_data;
restore datafile 13;
recover tablespace working_data;
recover datafile 13;
sql "alter tablespace working_data online";
sql "alter database datafile 13 online";
```

注意：
如果任何一个表空间含有有效的回滚段，这个恢复案例就有可能不能实现。在丢失了有效回滚段表空间的事件中，最好执行这个表空间或数据文件的脱机恢复操作。

案例 7：恢复丢失的未归档的联机重做日志

概况

今天不是 Bill 的好日子，外面是雷雨天，而 Bill 忘记了关闭汽车顶篷。更糟的是，闪电击中了 Bill 的数据中心，并且损坏了数据库所在的几个磁盘驱动器。

问题

在修复硬件后，Bill 发现丢失了所有联机重做日志以及一些数据文件。万幸的是，控制文件未受任何破坏。

解决方案

Bill 需要使用不完全恢复来还原数据库，由于无法使用联机重做日志，他不得不面对恢复操作会带来的数据丢失问题。

解决方案的实现

基于前面的这些考虑，Bill 设计和实现了下面的恢复计划：

(1) 确定副本中的最后一个归档日志。由于 Bill 要执行不完全恢复操作，所以他必须查询 RMAN，以确定已有备份中的最后一个归档日志，如下所示：

```
startup mount;
List backup of archivelog from time = 'sysdate-7';
```

查询输出与下面的输出大致相同，注意报告最后的日志序列号(日志序列号 3，以粗体表示)，它表示最后一个备份的归档的重做日志，这是 Bill 最远能够恢复到的位置。

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

```
BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ---------------
216     48K        DISK        00:00:03     16-AUG-09
        BP Key: 247 Status: AVAILABLE Tag: TAG20020816T095848
        Piece Name: D:\BACKUP\RECOVER\75E08R2P_1_1

  List of Archived Logs in backup set 216
  Thrd Seq     Low SCN    Low Time   Next SCN   Next Time
  ---- ------- ---------- --------- ---------- ---------
  1    2       1271924    16-AUG-09 1272223    16-AUG-09

BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ---------------
218     2K         DISK        00:00:02     16-AUG-09
        BP Key: 249 Status: AVAILABLE Tag: TAG20020816T100344
        Piece Name: D:\BACKUP\RECOVER\77E08RC1_1_1
  List of Archived Logs in backup set 218
  Thrd Seq     Low SCN    Low Time   Next SCN   Next Time
  ---- ------- ---------- --------- ---------- ---------
  1    3       1272223    16-AUG-09 1272321    16-AUG-09
```

(2) 使用现有的最后一个日志序列号，就可以执行下面的还原与恢复操作并打开这个数据库。Bill 首先使用 until sequence 参数还原数据库，这样可以确保所有数据库数据文件被还原到不晚于日志序列 3 的时间点，此外 force 参数能够确保还原所有数据文件。前面介绍过，时间点恢复操作的一个要求是所有数据库数据文件都必须被还原到一致的相同时间点。因此，至少将所有数据文件还原到 Bill 要恢复的时间点之前的一个时间点非常重要。

```
RMAN>restore database until sequence=4 thread=1 force;
```

(3) 如下所示，使用 until sequence 4 恢复数据库(由于 until sequence 恢复到列出序列号之前的序列号，所以 Bill 将最后一个序列号加 1，因此日志序列号为 4)：

```
RMAN>recover database until sequence=4 thread=1;
```

(4) 打开数据库：

```
RMAN>sql "alter database open resetlogs";
```

注意：
如果在联机重做日志丢失前通过 shutdown normal、immediate 或 transactional 命令正常地关闭了数据库，Bill 不需要恢复数据库就能打开这个数据库。

案例 8：使用 resetlogs 命令的恢复操作

概况

Bill 花费了一个晚上来执行恢复操作，然后他要求 Tim 监控还原操作和完成数据库

的恢复。一旦完成了恢复操作，Tim 就需要开始备份数据库。一个要求是在备份操作期间数据库是打开和有效的。

Tim 完成了恢复操作，随后，他使用 resetlogs 命令打开这个数据库(如案例 7 所示)。遵循业务需求，Tim 开始执行备份操作，同时允许用户访问数据库。

问题

但是，由于电源的不稳定，系统和数据库所在的磁盘驱动器再次被破坏。修复硬件后，Tim 发现丢失了一些数据文件，更糟的是，执行 resetlogs 命令后还没有备份完这些数据文件。幸运的是，该数据库在 3 个月前升级到了 Oracle Database 11g。因此，在 Oracle Database 10g 及其后版本中，Tim 可以更容易地通过 resetlogs 命令执行恢复操作。

当数据库升级到 Oracle Database 11g 时，LOG_ARCHIVE_FORMAT 参数被更改。DBA 添加了新的%R 参数，从而 LOG_ARCHIVE_FORMAT 现在如下所示：

```
SQL> show parameter log_archive_format
NAME                                 TYPE        VALUE
------------------------------------ ----------- -----------
log_archive_format                   string      ARC%S_%R.%T
```

因此，现在 LOG_ARCHIVE_FORMAT 参数包括%R 占位符，这意味着归档重做日志的文件名中包括 resetlogs ID。这为 DBA 执行恢复操作节省了大量时间。

解决方案

Tim 决定通过 resetlogs 命令使用 RMAN 恢复数据库，在 Oracle Database 10g 及其后版本中，这种恢复操作很容易就可以完成。

解决方案的实现

基于前面的这些考虑，Tim 设计和实现了下面的恢复计划：

(1) 加载数据库：

```
/u01>Rman target=/
RMAN>startup mount;
```

(2) Tim 知道数据文件 4 丢失，该数据文件是 USERS 表空间的一部分，因此他还原数据文件 4：

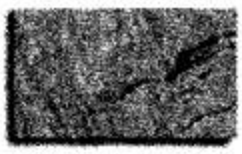
```
restore datafile 4;
```

注意：

Tim 甚至不需要复位数据库的对应物！这是 Oracle Database 10g 中的全新方法，它简化了通过对应物的恢复操作。

(3) 还原数据文件之后，Tim 现在恢复它，然后打开数据库：

```
recover datafile 4;
alter database open resetlogs;
```



注意：
您是否也采取了与 Tim 相同的方法：仅仅通过将数据文件脱机来打开数据库？Tim 然后可能还原数据文件，并且在恢复之前将该数据文件重新联机。下面是一些用于执行联机恢复的命令：

```
rman>startup mount
rman>sql 'alter database datafile 4 offline'
rman>alter database open;
rman>restore datafile 4;
rman>recover datafile 4;
rman>sql 'alter database datafile 4 online'
```

案例 9：手动完成失败的复制操作

概况

Tim 决定使用 RMAN 复制在不同的服务器上创建产品数据库的副本。Tim 通过运行 duplicate 命令将数据文件成功地还原到新的服务器上。这个产品数据库相当大，数据文件的还原进程花费了 6 个小时。

问题

Tim 忘记将归档日志移动到辅助服务器上用于介质恢复，因此复制操作失败。也就是说，数据库复制没有完全成功，这个数据库副本不具有新的 DBID。

解决方案

然而，Tim 并不担心，当然他也不会再花费 6 个小时来还原数据文件。如第 19 章所述，Tim 可以手动执行介质恢复操作，然后在数据库副本上使用 DBNEWID 实用程序创建新的 DBID，这样不需要 RMAN 的帮助就能够完成复制进程。

解决方案的实现

基于前面的这些考虑，Tim 设计和实现了下面的操作计划来完成失败的复制操作：

(1) 将归档日志从产品服务器移动到辅助服务器。

```
cd /space/oracle_user/OraHome1/oradata/sun92
tar -cvf arch.tar arch
gzip arch.tar
```

(2) 然后使用 FTP 将 arch.tar 文件移动到辅助系统，并且输入如下代码：

```
cd /space/oracle_user/OraHome1/oradata/sun92
gunzip arch.tar.gz
tar -xvf arch.tar
```

(3) 在数据库上执行手动的恢复操作。Tim 需要记下在目标数据库上有效的最后一个归档日志的序列号，然后设置恢复操作在这个序列号处停止。LOG_ARCHIVE_FORMAT 参数中的%s 变量表示日志序列号，并且会位于归档日志名中。通过在辅助站点上本地连

接辅助数据库，Tim 会从 SQL*Plus 执行下面的手动恢复操作：

```
ORACLE_SID=aux1
export ORACLE_SID
Sqlplus /nolog
SQL> connect / as sysdba
SQL> recover database using backup controlfile until sequence 11 thread 1;
```

(4) 使用 DBNEWID 实用程序为数据库副本创建新的 DBID。Tim 的辅助数据库虽然已被加载，但是还未被打开，这是使用 DBNEWID 实用程序的正确状态。如果不能确认辅助数据库的状态，您可以重新加载辅助数据库，这样不会有任何不良影响。

```
SQL> Shutdown immediate;
SQL> startup mount;
SQL> exit
$ nid target=/
$ sqlplus /nolog
SQL> connect / as sysdba
SQL> Shutdown immediate;
SQL> startup mount;
SQL> alter database open resetlogs;
```

案例 10：使用 RMAN 复制创建目标数据库的历史子集

概况

Svetlana 是出售玩具火车的站点的一位 DBA。她负责的产品数据库负荷很大，经常需要更新、插入和删除操作。一段时间后，她注意到针对特定库存跟踪表的数据采集操作性能减弱。

问题

Svetlana 怀疑这是由于不正确使用 Cost-Based Optimizer 造成的，并认为 Explain Plan 可能被改变。为了彻底检查问题，Svetlana 正寻找一种查看产品数据库子集历史快照的方法，她过去考虑过复制操作，但是在任何服务器上都没有克隆整个产品数据库的足够空间。

解决方案

Svetlana 可以使用 Oracle Database 11g 中在复制操作期间跳过某些表空间的新功能，通过这种方法，她可以在要测试的子集中只添加特定表空间。除了省略一些表空间之外，Svetlana 还可以在 duplication 命令中指定 until 子句来设置要测试的历史时间点。

解决方案的实现

第 19 章介绍了一个执行复制操作的 RMAN 实习。为了时间点复制子集能成功，Svetlana 必须在执行复制操作之前做好如第 19 章所述的所有准备工作。Svetlana 会在运行目标数据库的相同服务器上执行复制操作，因此需要确保使用了文件重命名策略。此外，她还必须以 NOMOUNT 模式启动辅助数据库，然后再运行复制操作代码。如下所示。

```
$ rman log=/space/backup/pitrdup.out
    connect target /
    connect auxiliary sys/password@aux1
    duplicate target database to aux1
    pfile=/space/oracle_user/OraHome1/admin/aux1/pfile/init.ora
    skip tablespace 'CWMLITE' , 'USERS' , 'ODM' , 'TOOLS'
    until sequence = 11 thread = 1
    logfile
    '/space/oracle_user/OraHome1/oradata/aux1/redo01.dbf' size 5m,
    '/space/oracle_user/OraHome1/oradata/aux1/redo02.dbf' size 5m,
    '/space/oracle_user/OraHome1/oradata/aux1/redo03.dbf' size 5m;
```

输出结果应当与下面的 RMAN 日志大致相同。为了节省篇幅，我们舍弃了一些不必要的内容，而突出显示了 RMAN 删除不必要的表空间的代码。需要记住的是，在 RMAN 中设置 until sequence 子句时，所指定的序列号不会被包含在恢复集中，因此在 Svetlana 的代码中，恢复操作是到归档日志序列号 10，而不是序列号 11。

```
Recovery Manager: Release 11.2.0.0.2 - Beta on Sun Aug 16 15:41:40 2009
    Starting Duplicate Db at 16-AUG-09
    using target database control file instead of recovery catalog
    allocated channel: ORA_AUX_DISK_1
    channel ORA_AUX_DISK_1: SID=280 device type=DISK
    Checking that duplicated tablespaces are self-contained
    Datafile 4 skipped by request
    Datafile 5 skipped by request

contents of Memory Script:
{
  set until scn 945053;
  set newname for datafile 1 to
"/home/oracle/app/oracle/oradata/aux1/system01.dbf";
  set newname for datafile 2 to
"/home/oracle/app/oracle/oradata/aux1/sysaux01.dbf";
  set newname for datafile 3 to
"/home/oracle/app/oracle/oradata/aux1/undotbs01.dbf";
  set newname for datafile 6 to
"/home/oracle/app/oracle/oradata/aux1/fbs01.dbf";
  restore
  clone database
  skip forever tablespace "USERS",
"RECO_CAT" ;
}
executing Memory Script
executing command: SET until clause
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
```

```
Starting restore at 16-AUG-09
executing script: Memory Script
executing command: SET until clause
executing command: SET NEWNAME
…
Starting recover at 16-AUG-09
datafile 3 not processed because file is offline
datafile 7 not processed because file is offline

starting media recovery
…
media recovery complete
Finished recover at 16-AUG-09
…
{ Alter clone database open resetlogs;}
executing script: Memory Script
database opened
printing stored script: Memory Script
{# drop offline and skipped tablespaces
sql clone "drop tablespace USERS including contents";
# drop offline and skipped tablespaces
sql clone "drop tablespace RECO_CAT including contents";

executing script: Memory Script
sql statement: drop tablespace USERS including contents
sql statement: drop tablespace RECO_CAT including content

Finished Duplicate Db at 16-AUG-09
Recovery Manager complete.
```

案例 11：在 ARCHIVELOG 模式中使用闪回恢复区中的映像副本恢复丢失的数据文件

概况

Tim 是一家大型制造公司的高级 DBA，他长期遵守这样的座右铭："不要因为您有狂想症就认为别人故意为难您。"由于坚持这种观点，Tim 一直在要求管理层增加存储设备，从而能够在磁盘上保存完整的数据库备份，而不是直接流向磁带。由于没有足够的空间用于磁盘上的每个数据文件备份，因此 Tim 选择一些具有代表性的数据文件，这些数据对于操作非常重要，并且最容易受到延长的停机时间的影响。

在将数据库升级到 11.2 版本之后，Tim 设置了他的闪回恢复区，并且在 FRA 中创建了最重要文件的映像副本。他现在获得一个每晚的增量备份，并且将该增量应用于这些文件的映像副本。

问题

在月末处理大量数据时，灾难不可避免地发生了。许多愤怒的部门负责人纷纷通过拨打寻呼机来找 Tim，而 Tim 关闭了电话并开始检查数据库。他发现一个关键的表空间

中有一个产生损坏的数据文件，文件号为 5。该文件在磁盘上只显示为 0 字节。只有恢复该文件，月末的处理工作才可以继续进行。

解决方案

Tim 切换到在 FRA 中运行的数据文件，在晚上之前，该数据文件是当前级别为 1 的增量备份。然后，只需要将归档日志应用于文件，并恢复到当前的时间点。

解决方案的实现

Tim 的准备工作意味着极大地减少了停机时间，下面是他采取的操作计划：

(1) 将数据文件 5 切换到 FRA 中的副本：

```
[oracle@dex oracle]$ rman target /
Recovery Manager: Release 11.2.0.0.0
connected to target database: V112 (DBID=2188245167)

RMAN> switch datafile 5 to copy;
using target database control file instead of recovery catalog
datafile 5 switched to datafile copy
"/u01/… /V102/datafile/o1_mf_payroll_21v9dhtp_.dbf"
```

(2) 恢复数据文件 5：

```
RMAN> recover datafile 5;

Starting recover at 16-AUG-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=124 devtype=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: sid=132 devtype=DISK

starting media recovery

archive log thread 1 sequence 208 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_208
1v9hwr2_.arc
archive log thread 1 sequence 209 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_209
1v9hzvb_.arc
archive log thread 1 sequence 210 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_210
1v9j3ts_.arc
archive log thread 1 sequence 211 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_211
1v9j4jz_.arc
archive log thread 1 sequence 212 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_212
1v9j59g_.arc
archive log thread 1 sequence 213 is already on disk as file
```

```
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_213
1v9jc07_.arc
archive log
filename=/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1
f_1_208_21v9hwr2_.arc thread=1 sequence=208
archive log
filename=
/…/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_209_21v9hzvb_.arc
thread=1 sequence=209
archive log
filename=
/…/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_210_21v9j3ts_.arc
thread=1 sequence=210
archive log
filename=
//flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_211_21v9j4jz_.arc
thread=1 sequence=211
media recovery complete, elapsed time: 00:00:03
Finished recover at 16-AUG-09
```

(3) 将数据文件 5 联机：

```
sql>alter database datafile 5 online;
```

案例 12：执行在闪回恢复区之外运行产品数据文件的恢复

概况

Tim 使用他的 FRA 设置极大地减少了由于数据文件 5 导致的停机时间。然而，现在他在 FRA 的外部运行产品数据库，替换了有问题的磁盘，并且现在准备将数据文件还原到正常的位置。

问题

当然，使用数据文件切换方法来减少 MTTR 是非常好的方法。但是，它意味着产品数据库在运行时带有位于 FRA 中的数据文件。这种情况可以持续一段时间，但是最终还是需要将数据文件切换回正确的、标准的位置。

解决方案

Tim 需要建立数据文件 5 的新备份，将其还原到初始的文件位置，然后临时地停机，同时切换回该数据文件并恢复它。

为了将产品数据文件还原到正确的产品环境，Tim 必须计划好临时停机时间。实现快速恢复的一种折衷方案就是这种预先计划的临时停机。但是，Tim 可以计划在深夜实施这种临时的停机，这时只有很少的用户会受到影响，并且完全可以最小化停机所需的时间。

解决方案的实现

Tim 将使用如下的操作计划。

(1) 使用数据文件 5 的新映像副本(作为选择，Tim 可以使用磁带中数据文件 5 的先前备份，但他认为相比于加载磁带并进行还原操作，使用文件的新备份并从磁盘中还原将更为快速)。

```
RMAN> backup as copy datafile 5 format
    '/u01/app/oracle/oradata/v102/payroll01.dbf';

    Starting backup at 16-AUG-09
    using channel ORA_DISK_1
    using channel ORA_DISK_2
    channel ORA_DISK_1: starting datafile copy
    input datafile fno=00005
    name=/u01/app/oracle/flash_recovery_area/V102/datafile/o1_mf_payroll_21v9dh
    p_.dbf
    output filename=/u01/app/oracle/oradata/v102/payroll01.dbf
    tag=TAG20060319T124728 recid=27 stamp=585492453
    channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
    Finished backup at 16-AUG-09
```

(2) 将数据文件 5 还原到初始的文件位置。在前面显示的代码中，文件直接被备份到初始位置，因此不需要进行还原。

(3) 将数据文件 5 切换到初始位置中的副本：

```
RMAN> switch datafile 5 to copy;

    datafile 5 switched to datafile copy
    "/u01/app/oracle/oradata/v102/payroll01.dbf"
```

(4) 还原数据文件 5 并将该文件联机：

```
RMAN> recover datafile 5;

    Starting recover at 16-AUG-09
    using channel ORA_DISK_1

    starting media recovery
    archive log thread 1 sequence 221 is already on disk as file
    /u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_221
    1v9x6vv_.arc
    archive log thread 1 sequence 222 is already on disk as file
    /u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_222
    1v9xbvk_.arc
    archive log thread 1 sequence 223 is already on disk as file
    /u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_223
    1v9xjl5_.arc
    archive log thread 1 sequence 224 is already on disk as file
    /u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_224
    1v9xk6m_.arc
    archive log
```

```
filename=/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1
f_1_221_21v9x6vv_.arc thread=1 sequence=221
media recovery complete, elapsed time: 00:00:02
Finished recover at 16-AUG-09
…
SQL> alter database datafile 5 online;

Database altered.

SQL> select name from v$datafile where file#=5;

/u01/app/oracle/oradata/v102/payroll01.dbf
```

案例 13：使用闪回数据库和介质恢复来指出使用 resetlogs 命令打开数据库的确切时间

概况

Farouk 没有注意到所有星期一和部分星期二早上发生的问题，因为这些问题小到不足以引起他的注意。最后，在一个星期二的早上，woodscrew 部门的一位经理打电话给 Farouk，表明 Woodscrew 数据库丢失了关于一些最流行的木螺钉模型的记录。Farouk 检查了数据库，确认是因为某个人删除了主表中的行。

问题

必须花费一定的时间来查找删除行的人以及删除的原因。该经理表明他在星期一的午餐期间注意到这个问题。Farouk 检查了他的闪回查询选项，但是发现该事务早于他的撤消段。他也检查了闪回数据库选项，并且发现他仍然可以闪回大约 48 小时。

解决方案

Farouk 将使用闪回数据库执行整个数据库的时间点恢复。此时没有其他的选择。然而，Farouk 并不知道确切的故障时间，因此他需要来回调整时间以确定故障发生之前执行的最靠后的事务。他将使用闪回数据库和归档日志恢复来前后滚动，直到找到打开数据库的正确时间。Farouk 将以只读模式打开数据库以检查表。

解决方案的实现

Farouk 设计了如下的计划：

(1) 闪回数据库到近似的第一个故障点：

```
RMAN> run {
2> flashback database to time "to_date('2006-03-19 13:30:00',
'YYYY-MM-DDHH24:MI:SS')";
3> }
Starting flashback at 19-MAR-06
allocated channel: ORA_DISK_1

starting media recovery
```

```
archive log thread 1 sequence 225 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_225
1vcwlnx_.arc
archive log thread 1 sequence 226 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_226
1vcwnlv_.arc
archive log thread 1 sequence 227 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_227
1vcwt8h_.arc
archive log thread 1 sequence 228 is already on disk as file
/u01/app/oracle/flash_recovery_area/V102/archivelog/2006_03_19/o1_mf_1_228
1vczqgf_.arc
…
media recovery complete, elapsed time: 00:00:24
Finished flashback at 16-AUG-09
```

(2) 选择受影响的表并查看值是否正确：

```
sql> alter database open read only;
sql> select count(*) from ws_app.woodscrew where thread_cnt=30;
```

(3) Farouk 发现缺少相应的值，因此他再次执行闪回：

```
sql> shutdown immediate;
sq> startup mount;

rman> run {
flashback database to time "to_date('2009-09-16 13:20:00',
'YYYY-MM-DD HH24:MI:SS')";}
```

(4) Farouk 再次检查值：

```
sql> alter database open read only;
sql> select count(*) from ws_app.woodscrew where thread_cnt=30;
```

(5) 现在有相应的值，但是缺少一些可以获得的行。因此，Farouk 再次关闭数据库，并且执行进一步的恢复。

```
sql> shutdown immediate;
sql> startup mount;

RMAN> recover database until sequence=227 thread=1;

sql> alter database open read only;
sql> select count(*) from ws_app.woodscrew where thread_cnt=30;
```

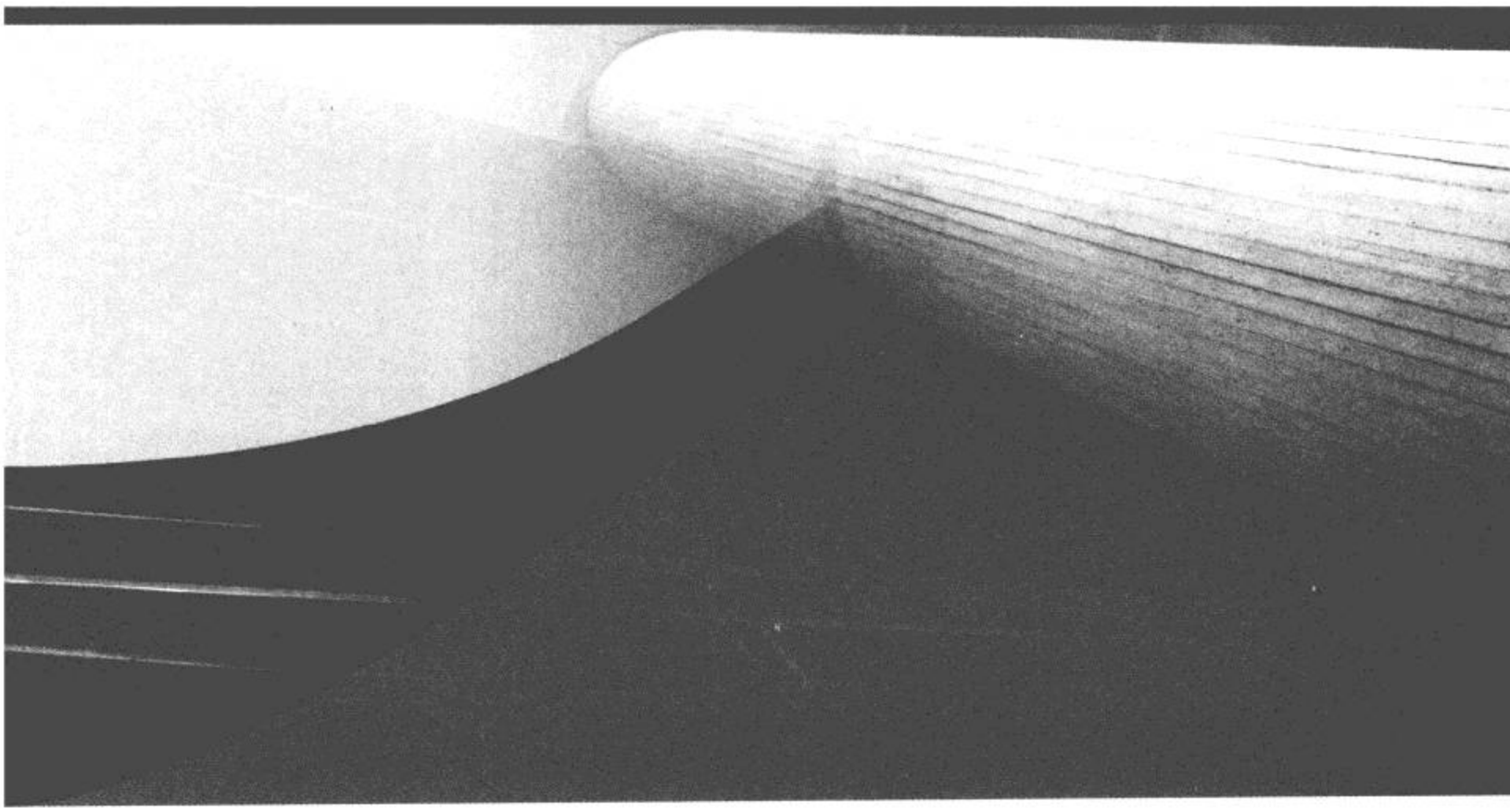
(6) 现在具有完整的值。Farouk 以读写模式打开数据库。

```
sql> shutdown immediate;
sql> startup mount;
```

```
sql> alter database open resetlogs;
```

23.3　小结

希望这些案例能够对您有所帮助。我们尽力为您提供了一些可能遇到的不同情况，并且提供了能够在测试系统上练习的解决方案，这样您就能够解决今后可能出现的各种问题。本章也是本书的最后一章，我们将 RMAN 备份与恢复的内容融汇在一起，以方便您的进一步学习。最后，我们希望您对本书的下一个版本提出建议。



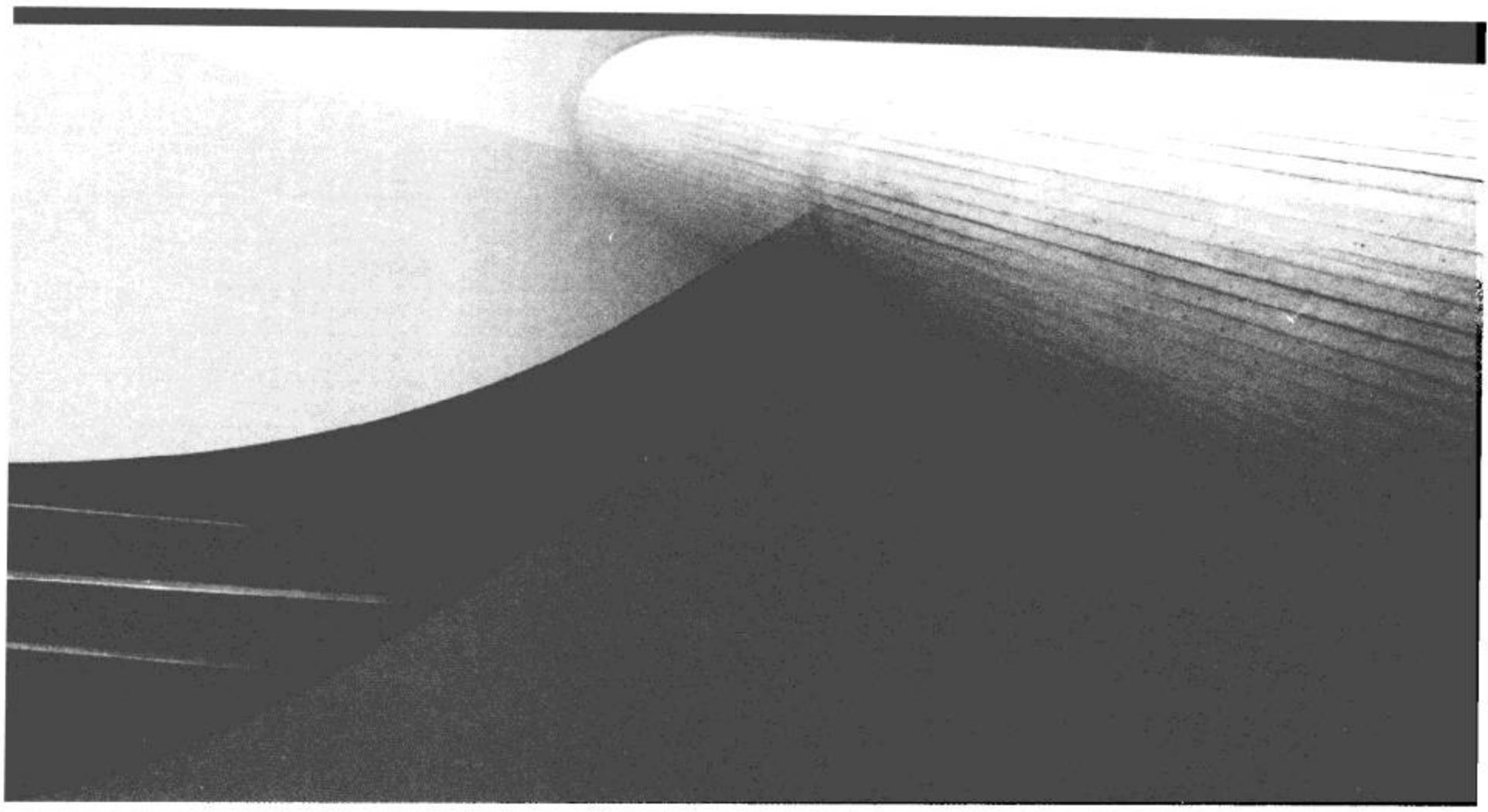
第V部分

附　　录

附录 A　RMAN 语法参考指南

附录 B　RMAN 脚本示例

附录 C　创建 RMAN 测试环境



附录 A

RMAN 语法参考指南

本附录是 RMAN 命令集的快速参考。针对每种 RMAN 命令，我们将简要介绍它的用途、相关的语法图、参考章节，以及使用该命令的一些示例。该附录也包含了 RMAN 保留字的列表。我们希望该附录能对您提供帮助，以方便您参考。

A.1 RMAN 保留字

表 A-1 提供了 RMAN 保留字的列表，在执行 RMAN 操作时应该避免使用这些保留字。

表 A-1 RMAN 保留字列表

,	#	(	)
/	{	}	\
<<<	>>>	;	&
_	'	=	^
@	.	:	abort
accessible	active	adviseid	Advise
AES128	AES192	AES256	Affinity
ater	algorithm	allocate	allow
All	alter	and	append
applied	archivelog	area	as
at	atall	autobackup	autoallocate
auxiliary	auxname	available	
backed	backuppiece	backupset	backups
backup	before	between	blockrecover
blocks	block	By	
cancel	catalog	change	channel
checksyntax	check	clear	clonename
clone	clone_cf	closed	cmdfile
command	comment	compatible	completed
compressed	compression	configure	connect
consistent	controlfilecopy	controlfile	Convert
copies	copy	corruption	create
critical	crosscheck	cumulative	current
database	datafile	datafilecopy	datapump
Days	dba	dbid	Db_file_name_convert
db_name	Db_recovery_file_dest	Db_unique_name	debug
decryption	default	Define	delete
deletion	destination	Detail	device
directory	diskratio	disk	Display
dorecover	drop	dump	Duplex
duplicate	duration		
echo	encryption	exclude	execute
exit	expired	export	

(续表)

failover	failure	filesperset	Files
final	flashback	force	foreign
forever	format	for	from
full			
get	global	grant	group
guarantee	g		
header	high	host	
Identified	identifier	id	immediate
import	inaccessible	incarnation	include
Including	incremental	input	instance
io			
Job			
kbytes	keep	krb	k
Level	libparm	library	like
limit	list	load	logfile
logical	logseq	logs	log
logscn	Low	logseq	
maintenance	mask	maxcorrupt	maxdays
maxopenfiles	maxpiecesize	maxseq	maxsetsize
maxsize	method	minimize	misc
mount	msglog	Msgno	M
names	name	need	New-line
Newname	new	nocatalog	nofau
nochecksum	nodevals	noduplicates	noexclude
nofilenamecheck	nofileupdate	nokeep	nologs
nomount	none	noparallel	noprompt
noredo	normal	not	no
null			
obsolete	offline	off	of
only	on	open	optimization
option	orphan	or	
packages	parallelism	parallelmediarestore	parallel
parameter	Parameter_value_conversion	parms	partial

(续表)

password	pfile	pipe	platform
plsql	plus	point	policy
pool	preview	primary	print
priority	privileges	proxy	put
Quit			
rate	rcvcat	rcvman	readonly
readrate	recall	recoverable	recovery
recover	redundancy	register	release
reload	remove	renormalize	repairid
repair	replace	replicate	report
resetlogs	reset	restart	restore
restricted	resync	retention	reuse
revoke	rpctest	rpc	run
save	schema	scn	script
section	send	sequence	setlimit
setsize	set	shipped	show
shutdown	since	size	skip
slaxdebug	snapshot	spfile	spool
sql	standby	startup	start
step	summary	switch	
tablespaces	tablespace	tag	target
tdes168	tempfile	test	thread
timeout	times	time	to
trace	transactional	transport	type
uncatalog	undo	unlimited	unrecoverable
unregister	until	upgrade	up
using			
validate	verbose	virtual	
window	with		

A.2　RMAN 命令列表

表 A-2 列出了每个 RMAN 命令。在接下来的 A.4 节中将介绍每个命令的细节。

表 A-2　RMAN 命 令

@	@@	ADVISE FAILURE	ALLOCATE CHANNEL
ALLOCATE CHANNEL FOR MAINTENANCE	BACKUP	CATALOG	CHANGE
CONFIGURE	CONNECT	CONVERT	CREATE CATALOG
CREATE SCRIPT	CROSSCHECK	DELETE	DELETE SCRIPT
DROP CATALOG	DROP DATABASE	DUPLICATE	EXECUTE SCRIPT
EXIT	FLASHBACK DATABASE	GRANT	HOST
IMPORT CATALOG	LIST	PRINT SCRIPT	QUIT
RECOVER	REGISTER DATABASE	RELEASE CHANNEL	REPAIR FAILURE
REPLACE SCRIPT	REPORT	RESET DATABASE	RESTORE
RESYNC CATALOG	REVOKE	RMAN	RUN
SEND	SET	SHOW	SHUTDOWN
SPOOL	SQL	STARTUP	SWITCH
TRANSPORT TABLESPACE	UNREGISTER	UPGRADE CATALOG	VALIDATE

A.3　RMAN 说明符和操作数列表

表 A-3 提供了各种 RMAN 说明符和操作数的名称。在本章后面的 A.5 节中将详细介绍每个说明符和操作数。

表 A-3　RMAN 说明符和操作数列表

allocOperandList	archivelogRecordSpecifier	completedTimeSpec
connectStringSpec	datafileSpec	deviceSpecifier
fileNameConversionSpec	forDbUniqueNameOption	foreignLogRecordSpecifier

(续表)

formatSpec	keepOption	listObjList
maintQualifier	maintSpec	obsOperandList
recordSpec	tempfileSpec	untilClause

A.4 RMAN 命令列表语法细节

本节提供了每个 RMAN 命令的详细信息。对于每个 RMAN 命令，您将可以看到如下内容：

- 命令的概述
- 命令的语法图
- 本书中可以找到该命令更多细节的相关章节
- 该命令的使用示例

A.4.1 @命令

@命令允许用户从RMAN或另一个RMAN命令文件中执行某个命令文件(RMAN脚本)，如图 A-1 所示。

语法图

图 A-1 语法图

参考章节

第 16 章提供了更多有关该命令用法的信息。

示例

```
@/usr/oracle/rman/script/rmanbackup.scr
```

A.4.2 @@命令

@@命令允许用户从 RMAN 或另一个 RMAN 命令文件中执行某个命令文件(RMAN脚本)，如图 A-2 所示。如果使用@@命令，则从与调用脚本相同的位置执行被调用的脚本。

语法图

图 A-2　语法图

参考章节

第 16 章提供了更多有关该命令的信息。

示例

```
@@rmanbackup2.scr
```

A.4.3　advise failure 命令

advise failure 命令用于显示对特定错误的各类修复选项。 advise failure 命令可显示由数据恢复顾问(DRA)发现的所有错误，如图 A-3 所示。

语法图

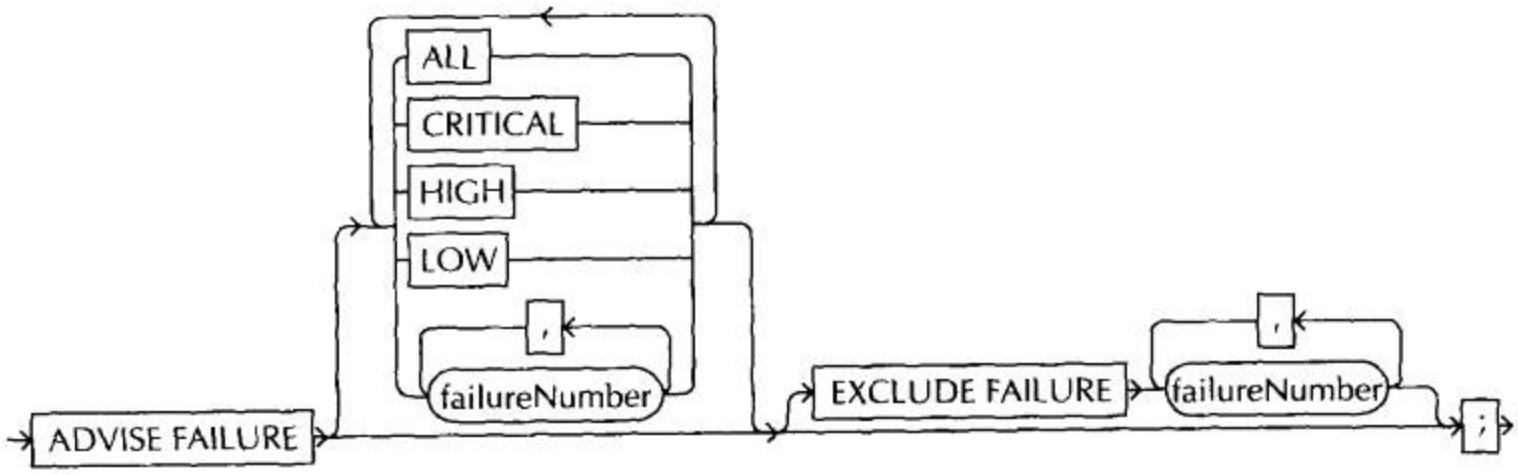


图 A-3　语法图

参考章节

第 12 章提供了更多的关于该命令的信息。

示例

```
advise failure;
```

A.4.4　allocate channel 命令

allocate channel 命令允许用户建立信道。信道表示 RMAN 与数据库实例之间的连接，尽管信道是预先配置，但对于任何 RMAN 备份、还原或维护操作来说，都必须至少分配一个信道，如图 A-4 所示。

语法图

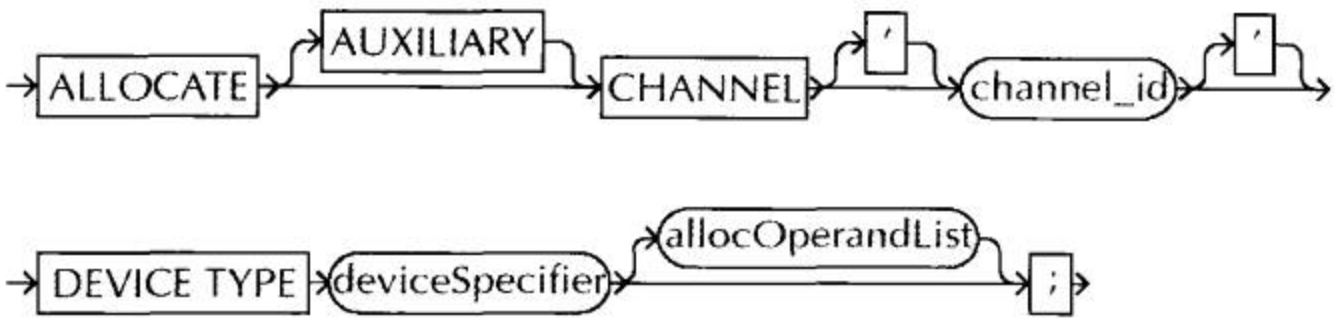


图 A-4　语法图

参考章节

在本书中，许多章节中都引用了该命令。第 3 章讨论了自动信道配置，从而在多数情况下不需要使用 allocate channel 命令。第 11、12、14 章专门介绍了使用 allocate channel 命令的详细信息。

示例

```
allocate channel c1 device type disk format '/u01/orabackup/robprod/
robprod_%U';

allocate channel t1 DEVICE TYPE 'SBT_TAPE' PARMS
'ENV=(TDPO_OPTFILE=/opt/oraback/rman/delta_${ORACLE_SID}.opt)'
RATE 1 G FORMAT '${ORACLE_SID}_%U' MAXPIECESIZE 20 G MAXOPENFILES 64;
```

A.4.5　allocate channel for maintenance 命令

在准备使用 change、crosscheck 或 delete 命令前，需要使用 allocate channel for maintenance 命令分配信道。如果配置了自动信道，就不需要使用该命令，如图 A-5 所示。

语法图

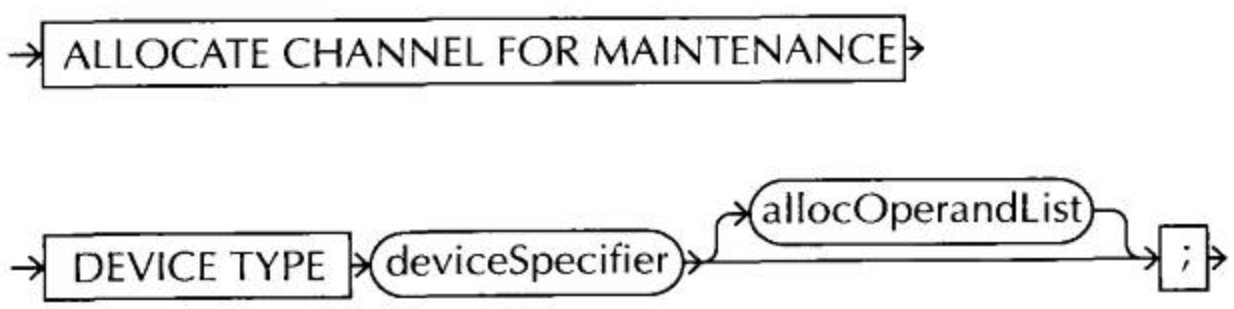


图 A-5　语法图

参考章节

本书中没有提供对 allocate channel for maintenance 命令的介绍，因为 Oracle 建议最好使用自动信道分配。

示例

```
allocate channel for maintenance device type sbt;
```

A.4.6　alter database 命令

alter database 命令允许加载或打开 RMAN 连接的目标数据库，如图 A-6 所示。

语法图

图 A-6　语法图

参考章节

在第 11、12、和 14 章中都示例说明了 alter database 命令。

示例

```
alter database open resetlogs;
```

A.4.7　backup 命令

backup 命令用于备份数据库，该命令支持联机和脱机备份，并且也支持数据库文件、数据库文件副本、归档日志和备份集的备份，如图 A-7～图 A-18 所示。

语法图

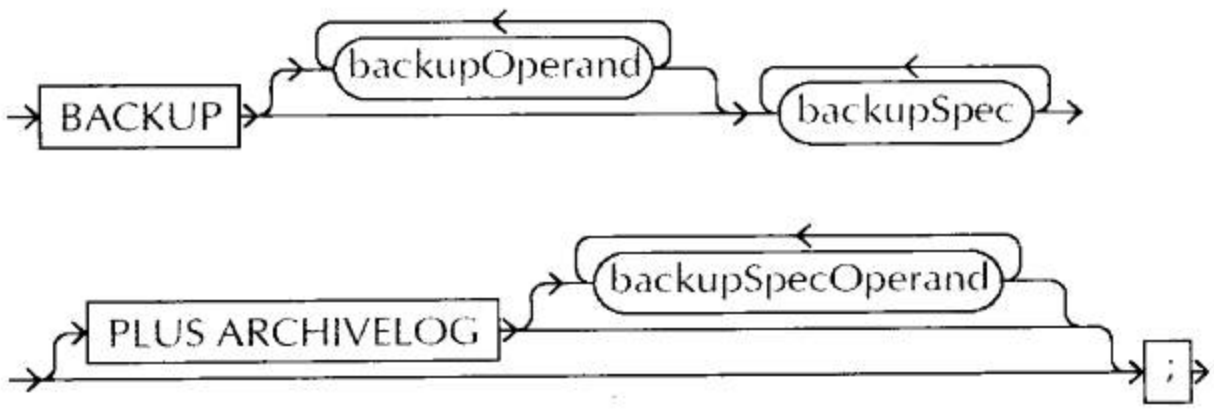


图 A-7　语法图

backupOperand 参数语法图

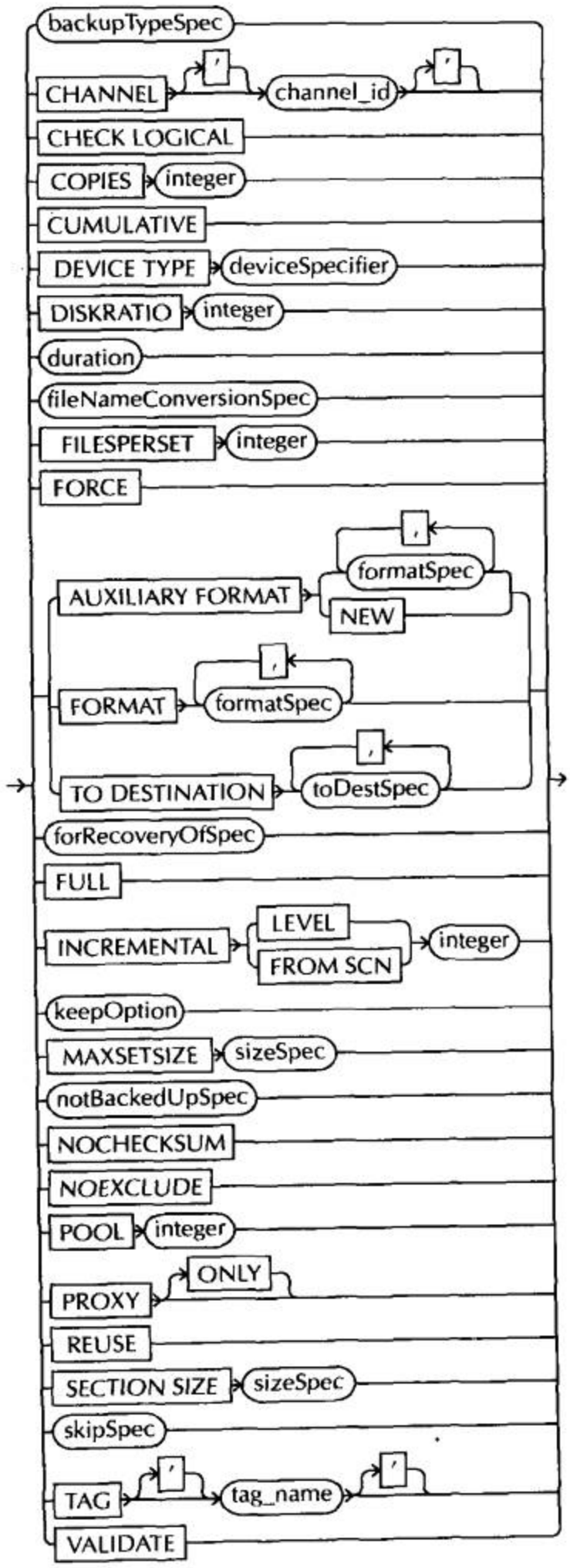


图 A-8 语法图

backupSpec 参数语法图

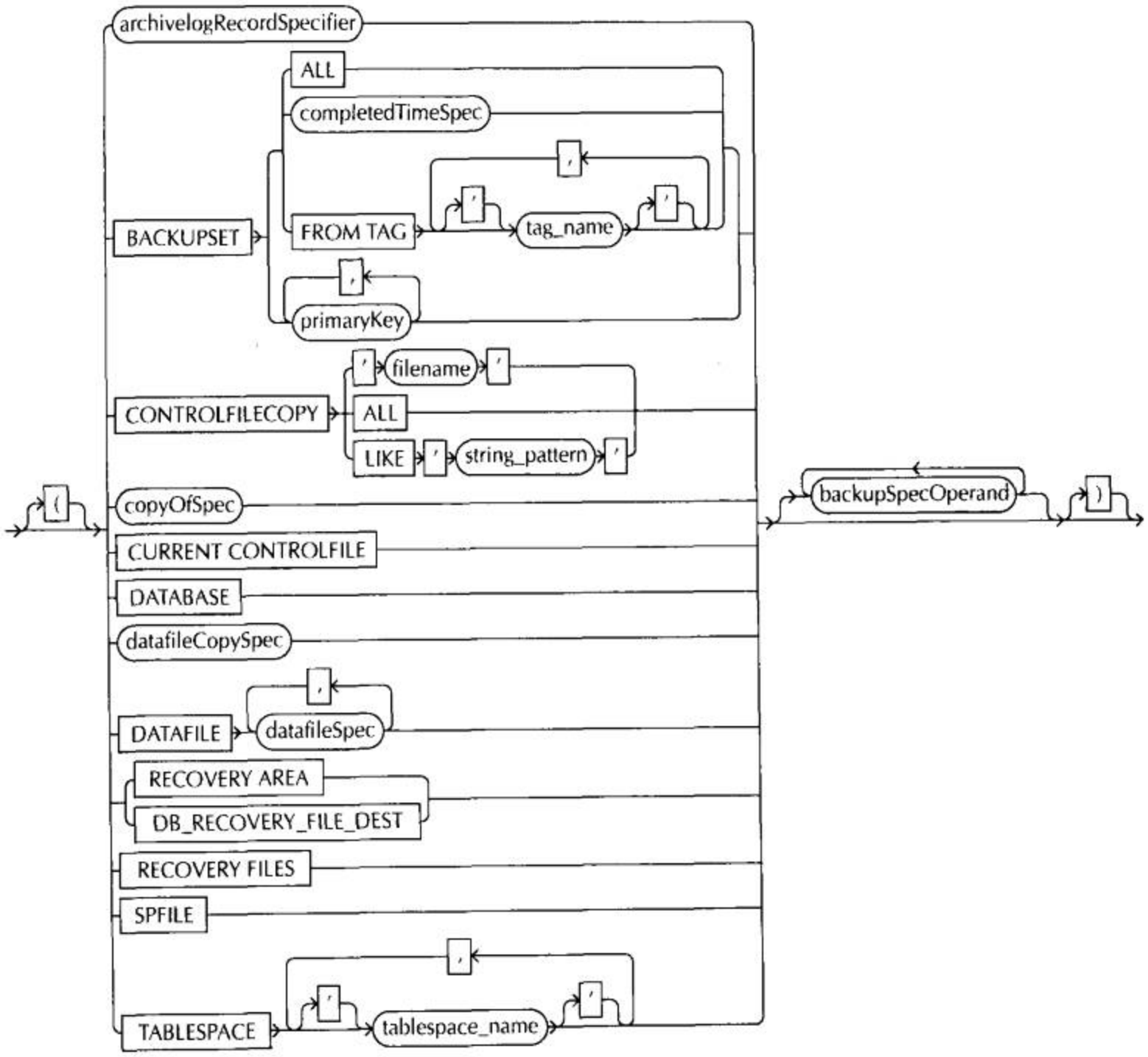


图 A-9　语法图

backupSpecOperand 参数语法图

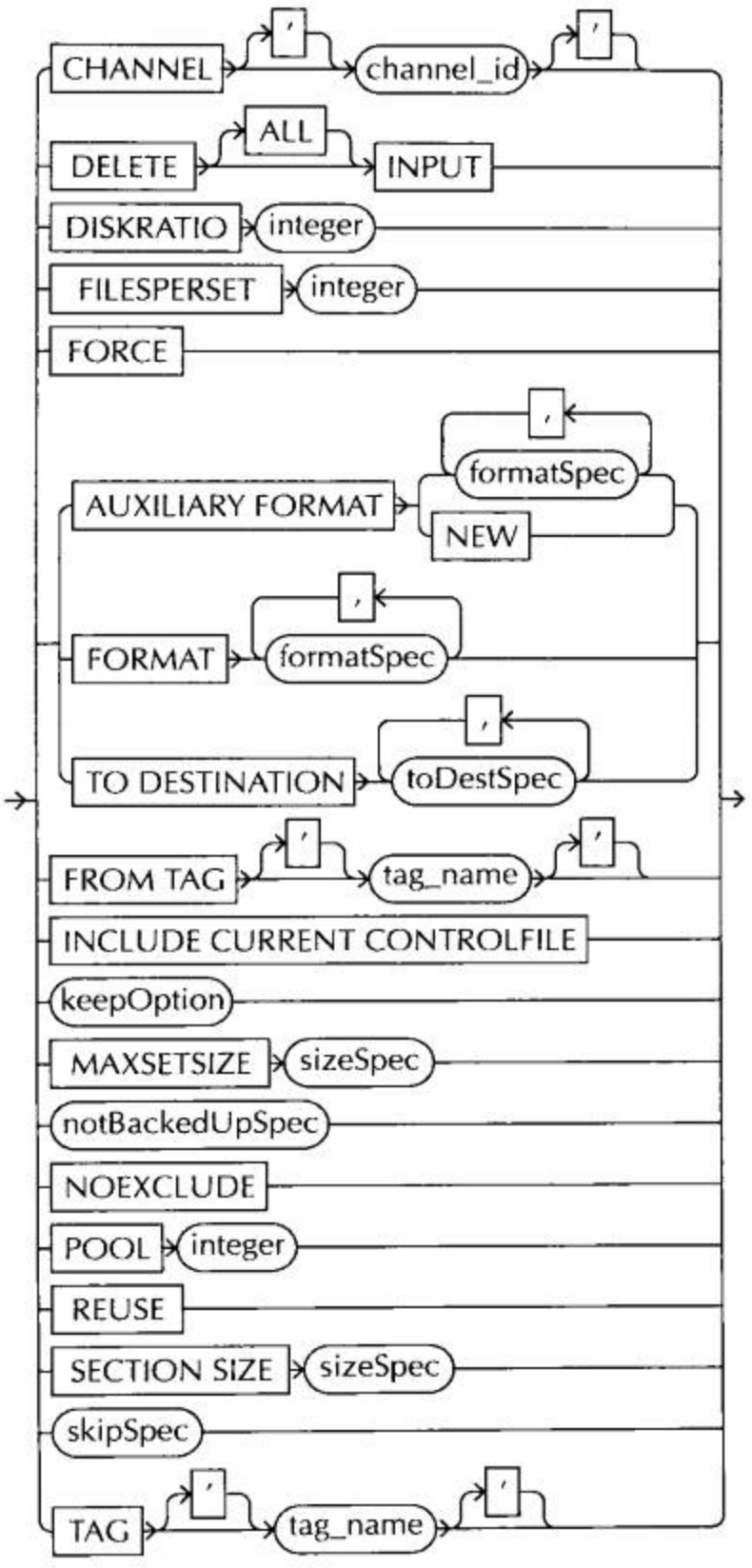


图 A-10　语法图

backupTypeSpec 参数语法图

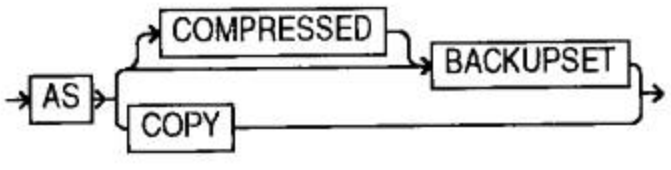


图 A-11　语法图

copyOfSpec 参数语法图

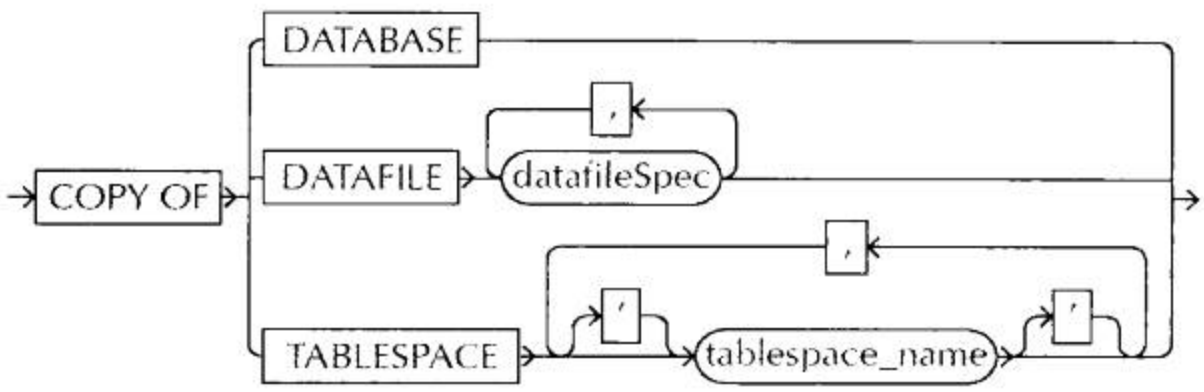


图 A-12　语法图

datafileCopySpec 参数语法图

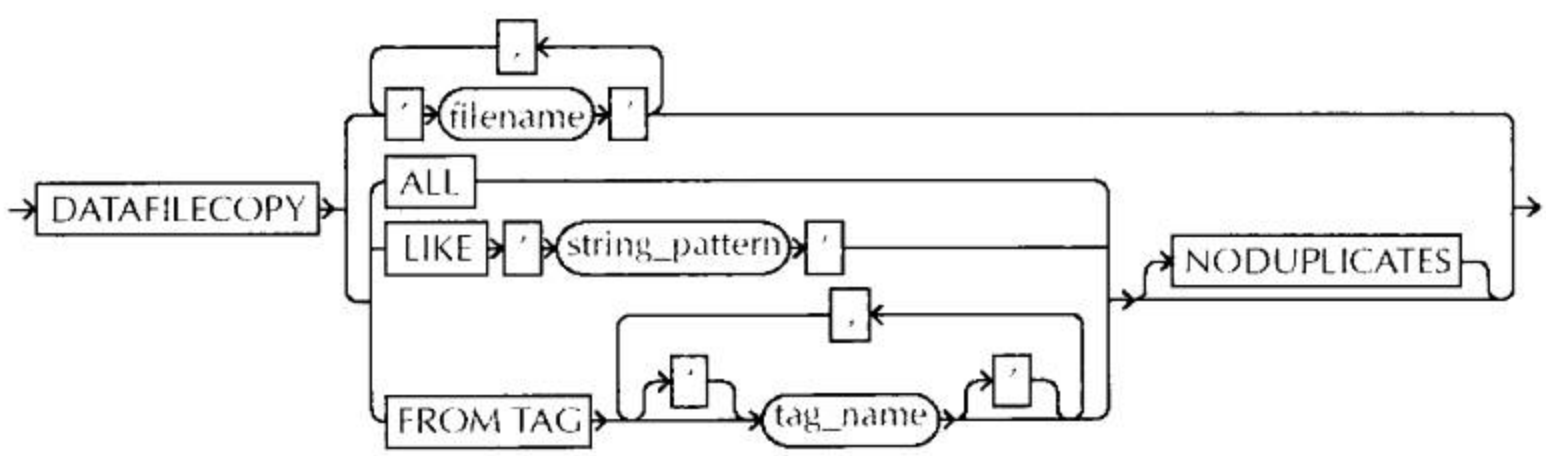


图 A-13　语法图

duration 参数语法图

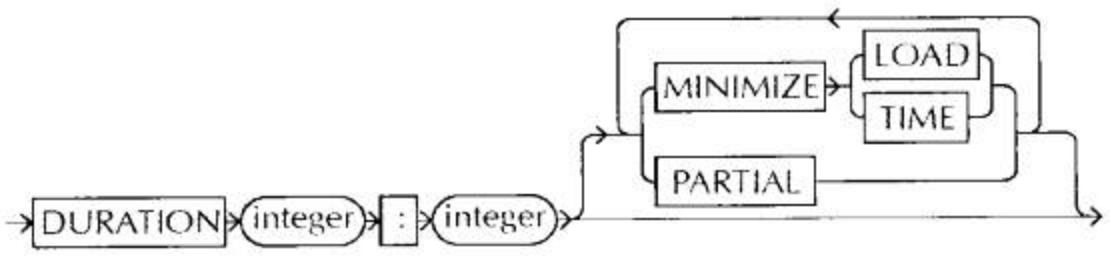


图 A-14　语法图

forRecoverOfSpec 参数语法图

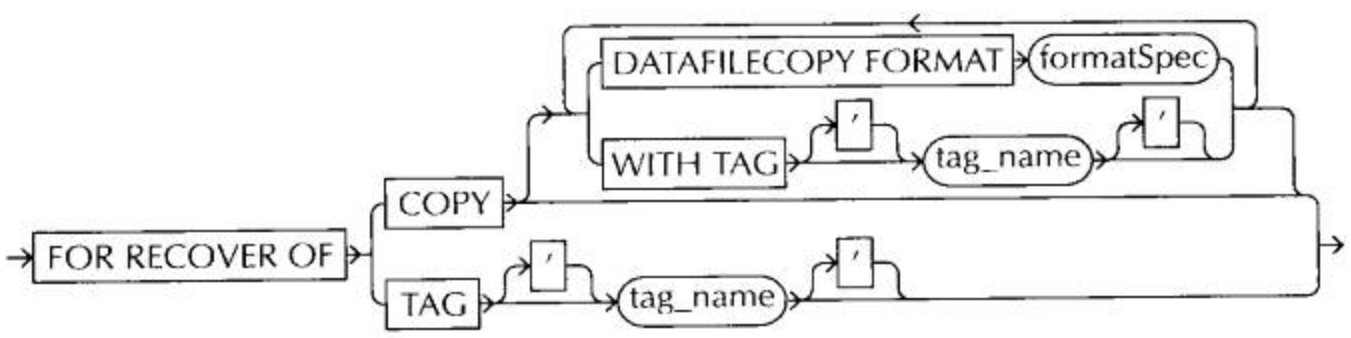


图 A-15　语法图

notBackedUpSpec 参数语法图

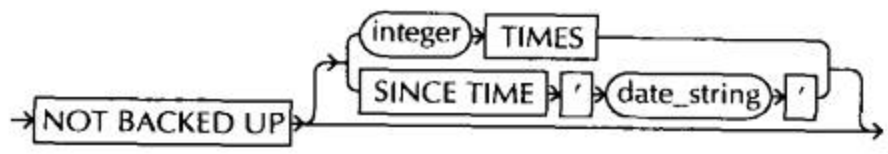


图 A-16 语法图

sizeSpec 参数语法图

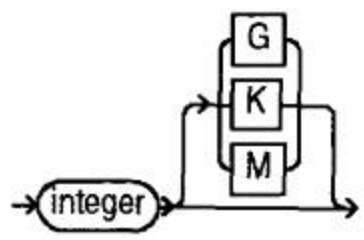


图 A-17 语法图

skipSpec 参数语法图

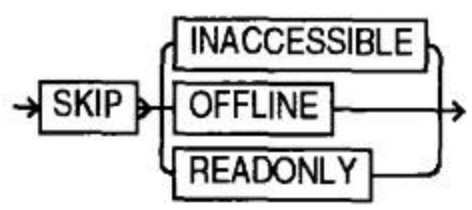


图 A-18 语法图

参考章节

第 11 章详细介绍了 backup 命令。

示例

```
backup database plus archivelog delete input;
backup tablespace users, system;
backup as copy datafile 1,2,3;
backup incremental level=0 check logical database plus archivelog;
```

A.4.8 catalog 命令

catalog 命令使用户能够完成下面这些功能，如图 A-19 所示：

- 将与备份片和映像副本有关的元数据添加入数据库恢复目录和控制文件。
- 在闪回恢复区(FRA)中对有效的备份集、数据文件副本和归档重做日志编写目录。
- 在 ASM、OMF 和用户定义的磁盘位置中对有效的备份编写目录。
- 在 RMAN 存储库中将数据文件副本作为 0 级备份记录，这样允许用户在增量备份资源中使用这个数据文件副本。

语法图

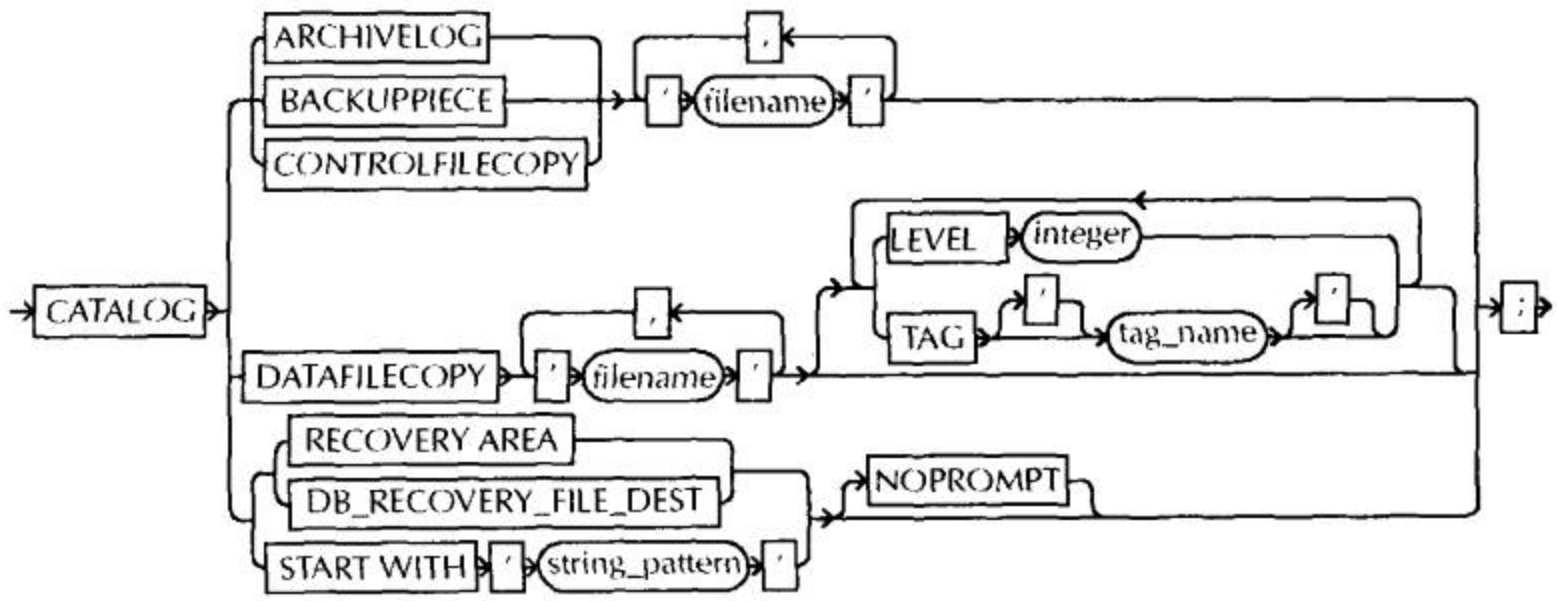


图 A-19　语法图

参考章节

第 16 章详细介绍了 catalog 命令的用法。

示例

```
catalog recovery area noprompt;

catalog datafilecopy 'd:\oracle\oraback\system01.bak'
'd:\oracle\oraback\users01.bak' level 0;
```

A.4.9　change 命令

change 命令用于将 RMAN 中的备份片、映像副本和归档的重做日志的状态从 UNAVAILABLE 状态改为 UNAVAILABLE 状态。此外，该命令可以完全删除储存库中的备份或副本，并且可用于修改对于给定备份或副本有效的保留策略，如图 A-20～图 A-24 所示。

语法图

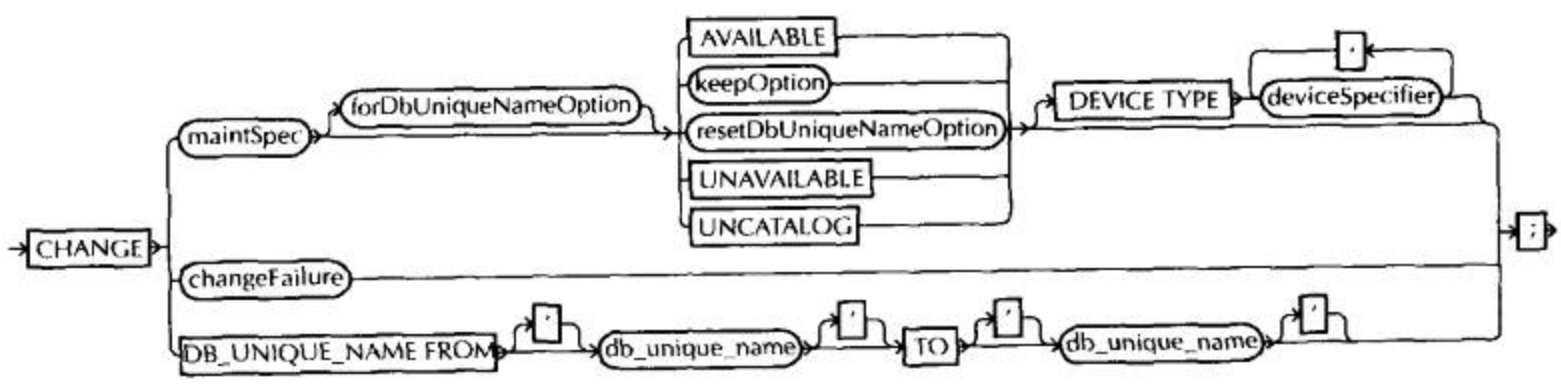


图 A-20　语法图

maintSpec 参数语法图

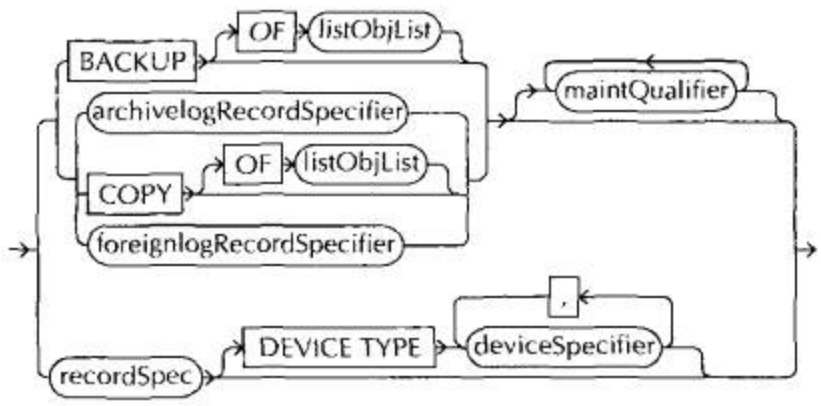


图 A-21　语法图

forDbUniqueNameOption 参数语法图

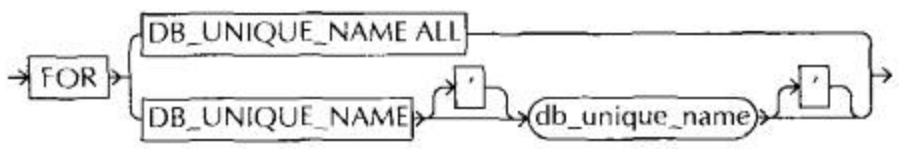


图 A-22　语法图

resetDbUniqueNameOption 参数语法图

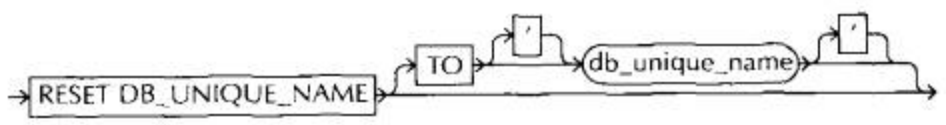


图 A-23　语法图

changeFailure 参数语法图

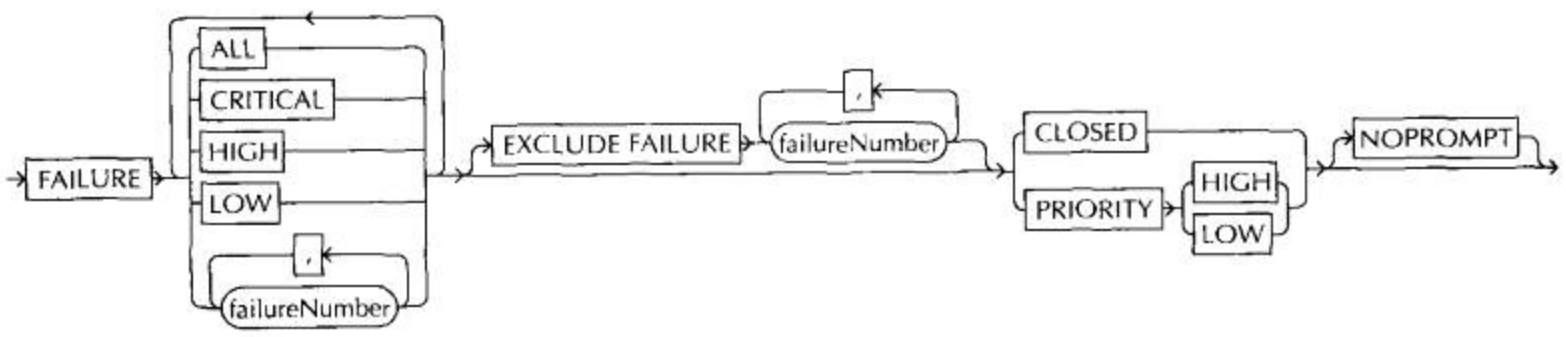


图 A-24　语法图

参考章节

第 16 章讨论了 change 命令的用法。

示例

```
change backup tag 'Gold_Backup' keep forever nologs;
```

A.4.10　configure 命令

configure 命令用于配置持久的 RMAN 配置设置，这些设置应用于所有的 RMAN 会话，除非修改或禁用它们。这些设置保存在数据库控制文件中，如图 A-25～图 A-30 所示。

语法图

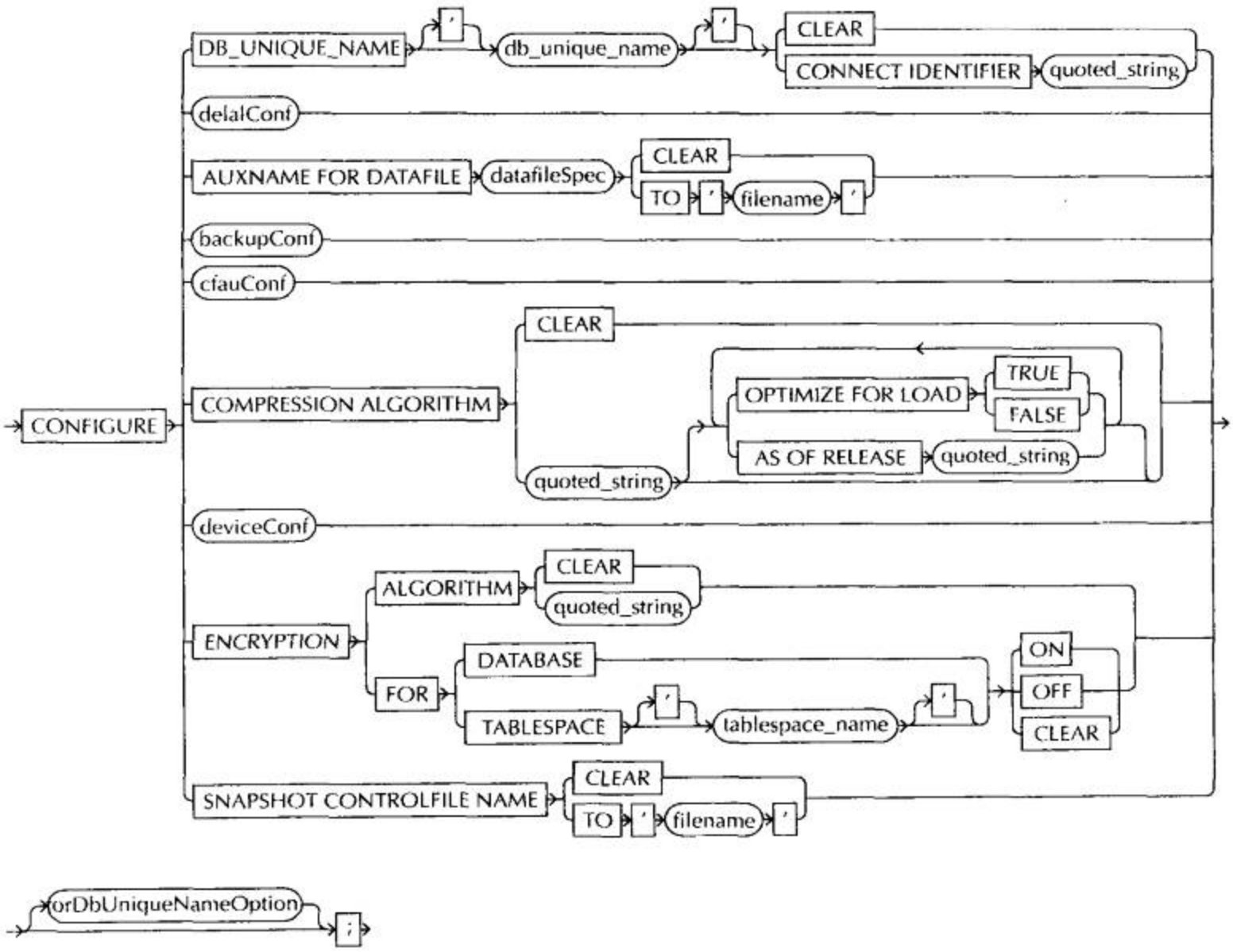


图 A-25　语法图

delal Conf 参数语法图

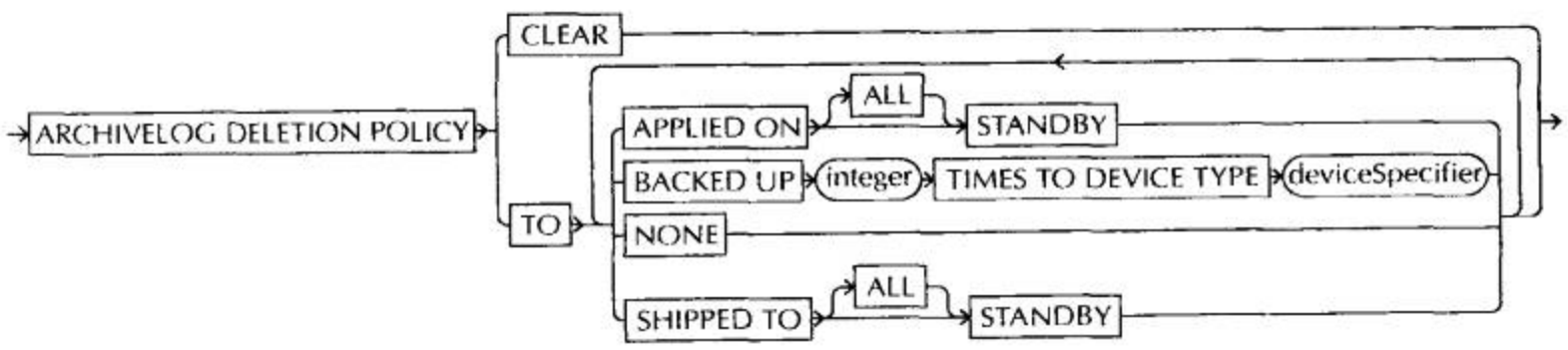


图 A-26　语法图

backupConf 参数语法图

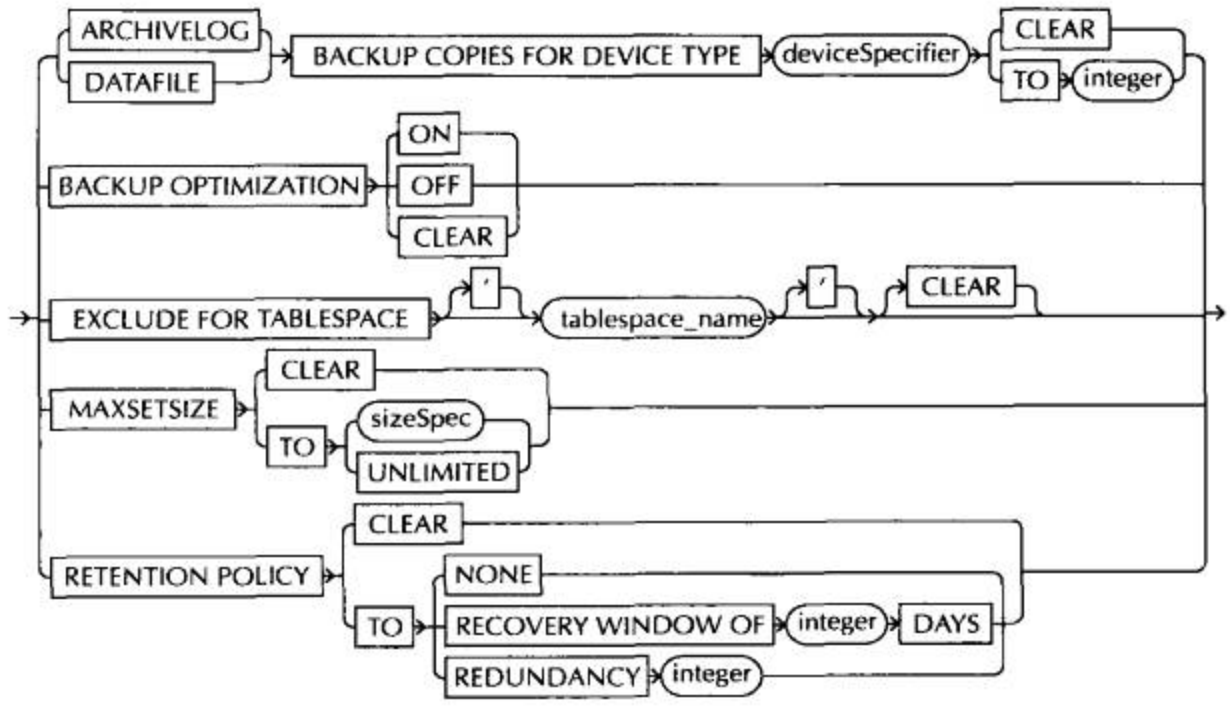


图 A-27　语法图

cfauConf 参数语法图

图 A-28　语法图

deviceConf 参数语法图

图 A-29　语法图

forDbUniqueNameOption 参数语法图

图 A-30　语法图

参考章节

第 3 章非常详细地介绍了 configure 命令。

示例

```
configure retention policy to recovery window of 180 days;
configure backup optimization off;
configure default device type to 'sbt_tape';
configure controlfile autobackup on;
configure controlfile autobackup format for
device type 'sbt_tape' ''%f'';
configure controlfile autobackup format for device type disk to
'/oradata/oraback/rman/robone/%f';
configure device type 'sbt_tape' parallelism 4;
configure device type disk parallelism 2;
configure datafile backup copies for device type 'sbt_tape' to 1;
configure datafile backup copies for device type disk to 1;
configure archivelog backup copies for device type 'sbt_tape' to 1;
configure archivelog backup copies for device type disk to 1;
configure channel device type 'sbt_tape' parms
'env=(tdpo_optfile=/opt/oracle/rman/tdpo_a273.opt)'
rate 9 g format 'a273_%u' maxpiecesize 20 g maxopenfiles 64;
configure channel device type disk format
'/oradata/orabck01/rman/a273/ora_a273_df%t_s%s_s';
configure auxiliary channel device type 'sbt_tape' parms
'env=(tdpo_optfile=/opt/oracle/rman/tdpo_a273_dup.opt)'
rate 99 g format 'a273_%u' maxpiecesize 20 g maxopenfiles 64;
configure maxsetsize to 20 g;
configure snapshot controlfile name to
'/oradata/orabck01/rman/a273/snapcf_a273.f';
configure channel 1 device type sbt
connect 'sys/robert@test1' parms 'env=(nsr_server=bktest1)';
```

A.4.11　connect 命令

connect 命令用于从 RMAN 提示符中连接到目标数据库、恢复目录或辅助实例，如图 A-31 所示。

语法图

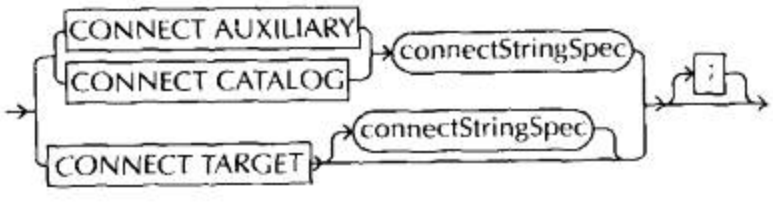


图 A-31　语法图

参考章节

几乎每一章都有 connect 命令的示例。

示例

```
connect target sys/Robert@maindb
connect catalog rman/rman@rmancat
```

A.4.12　convert 命令

当在不同平台之间移动表空间和数据库时，使用 convert 命令。如果源平台和目标平台具有不同的字节顺序格式(尾数)，就必须使用 convert 命令来使数据文件符合新平台的要求。convert 命令也可用于将数据文件移入和移出 ASM 存储，如图 A-32～图 A-35 所示。

语法图

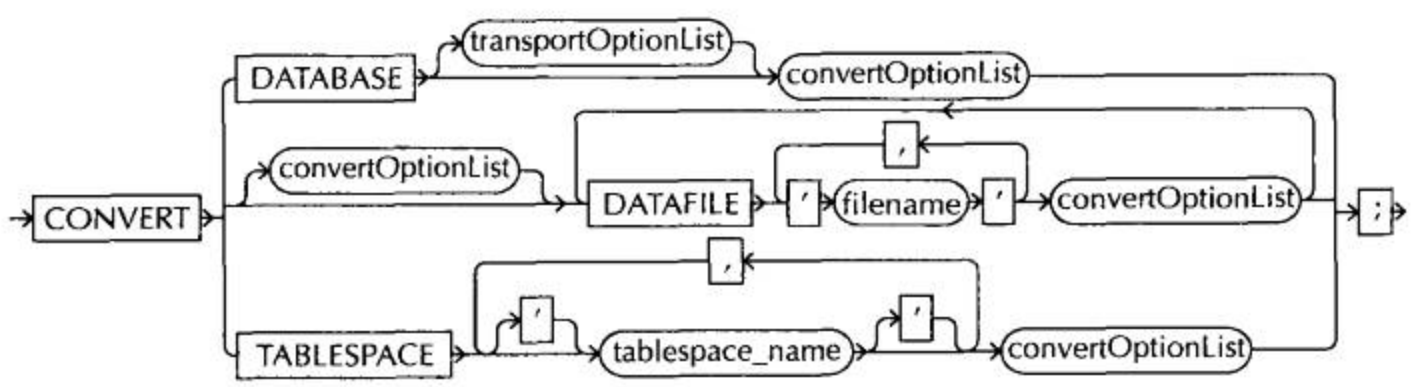


图 A-32　语法图

transportOptionList 参数语法图

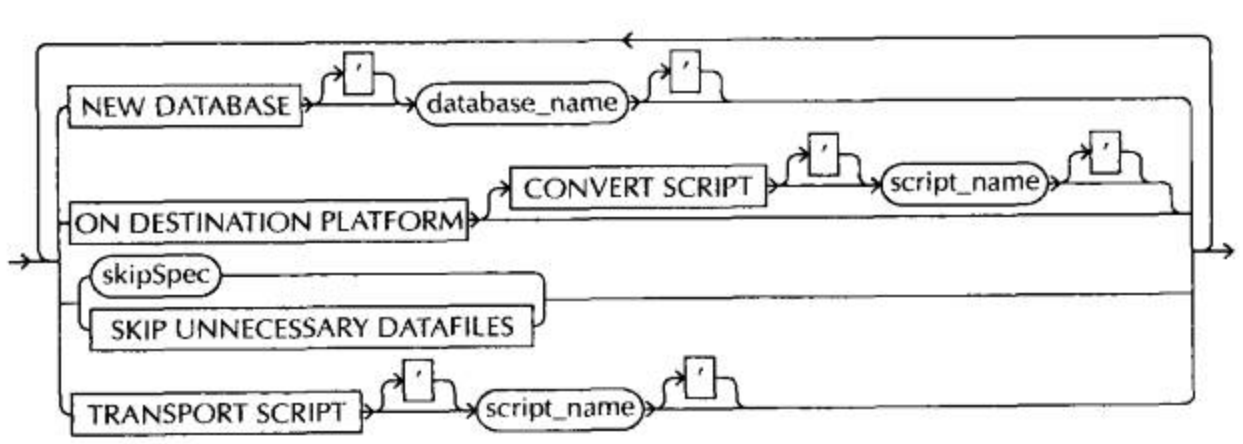


图 A-33　语法图

skipSpec 参数语法图

图 A-34　语法图

convertOptionList 参数语法图

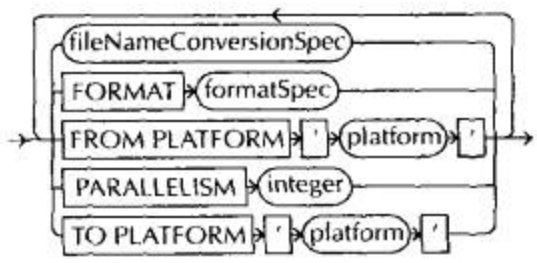


图 A-35　语法图

参考章节

第 14 章详细介绍了 convert 命令。

示例

```
-- Copy regular datafiles into ASM storage
convert datafile '/disk1/oracle/dbs/my_tbs_f1.df',
'/disk1/oracle/dbs/t_ax1.f'
format '+asmfile';

-- Copy a tablespace from ASM storage to disk.
convert tablespace tbs_2 format '/tmp/tbs_2_%U.df';

-- Convert a database, moving it to a new platform
convert database new database 'convdb'
transport script '/ora01/scripts/transport/transportscript'
to platform 'Microsoft Windows IA (32-bit)'
db file name convert '/ora01/oracle/dbs' '/newora/converted';
```

A.4.13　create catalog 命令

create catalog 命令用于创建恢复目录模式，如图 A-36 所示。

语法图

图 A-36　语法图

参考章节

第 10 章介绍了 create catalog 命令。

示例

```
create catalog tablespace rec_cat;
```

A.4.14　create script 命令

create script 命令允许用户在 RMAN 中创建存储的命令脚本。使用该命令时，必须使用恢复目录，如图 A-37 所示。

语法图

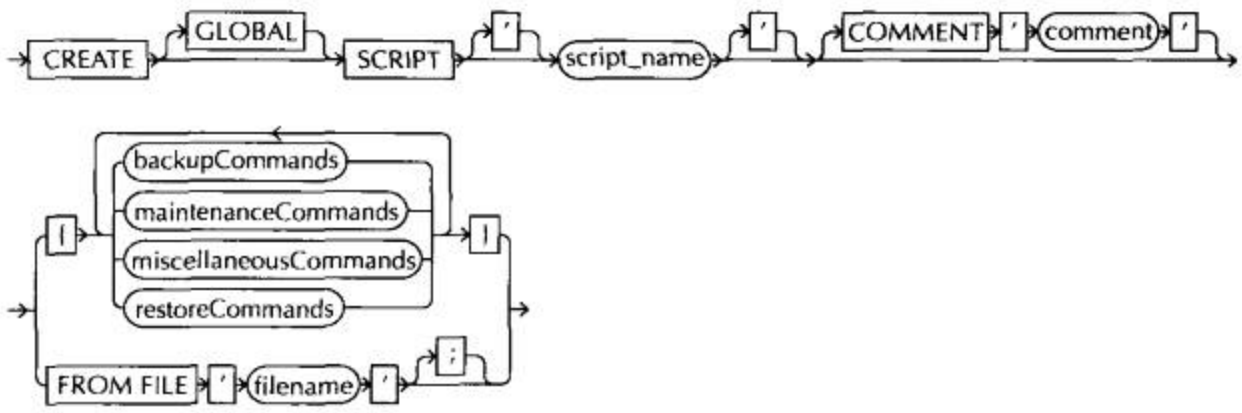


图 A-37　语法图

参考章节

第 10 章介绍了 create script 命令。

示例

```
create script backup_db {backup database plus archivelog delete input;}
```

A.4.15　crosscheck 命令

crosscheck 命令用于验证在备份介质上的控制文件或恢复目录中列出的已有备份集、备份集片。备份介质上没有的任何物理备份文件(备份集片、副本等)被标记为 EXPIRED，并且 RMAN 不会使用其进行恢复操作。如果备份文件在后面出现，则在随后的 crosscheck 操作中将其标记为 AVAILABLE，如图 A-38～图 A-39 所示。

语法图

图 A-38　语法图

maintSpec 参数语法图

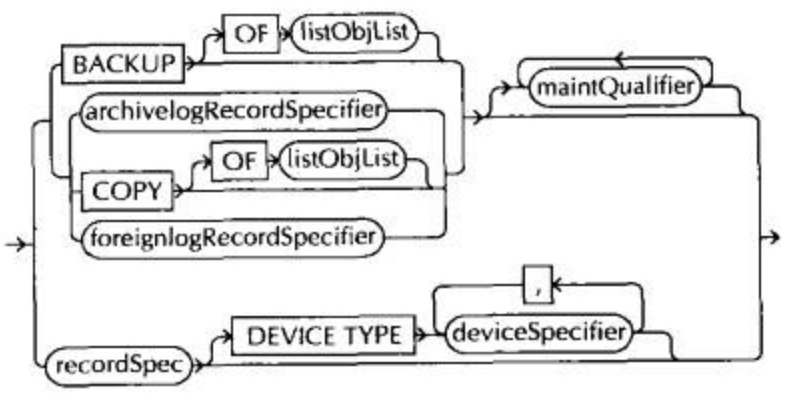


图 A-39　语法图

参考章节

第 16 章介绍了 crosscheck 命令。

示例

```
crosscheck backup;
crosscheck backup completed between 'sysdate' and 'sysdate-30';
crosscheck archivelog all;
crosscheck archivelog time between 'sysdate' and 'sysdate-30';
```

A.4.16　delete 命令

delete 命令用于将恢复目录或控制文件中的备份和副本标记为 DELETED 状态。备份或副本一旦被标记为 DELETED 状态，其状态就不可更改。可以使用 delete 将标记为 EXPIRED(通过使用 crosscheck 命令)或 OBSOLETE(通过预先定义的保留标准)的对象标记为 DELETED，如图 A-40～图 A-42 所示。

语法图

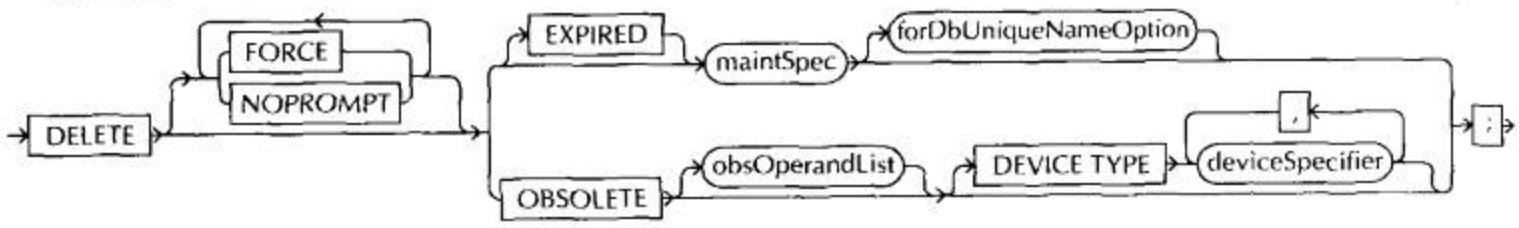


图 A-40　语法图

maintSpec 参数语法图

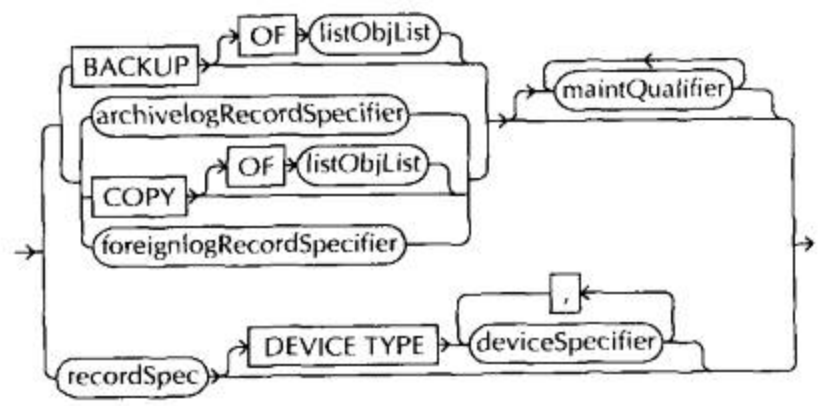


图 A-41　语法图

forDBUniqueNameOption 参数语法图

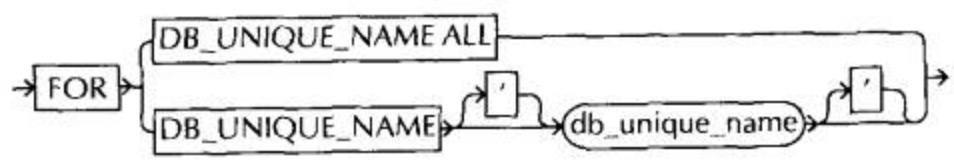


图 A-42　语法图

参考章节

第 16 章介绍了 delete 命令。

示例

```
delete expired backup;
```

```
delete expired archivelog;
```

A.4.17 delete script 命令

delete script 命令允许用户删除在恢复目录中存储的已有 RMAN 脚本，如图 A-43 所示。

语法图

图 A-43 语法图

参考章节

第 10 章详细介绍了 *delete script* 命令。

示例

```
delete script backup;
```

A.4.18 drop catalog 命令

drop catalog 命令用于删除恢复目录模式，如图 A-44 所示。

语法图

图 A-44 语法图

参考章节

第 10 章介绍了 drop catalog 命令的有关内容。

示例

```
drop catalog;
```

A.4.19 drop database 命令

drop database 命令用于删除目标数据库。如果连接到恢复目录，则也将取消注册数据库。当删除数据库时，RMAN 将删除属于该目标数据库的所有数据文件、联机日志和控制文件，如图 A-45 所示。

语法图

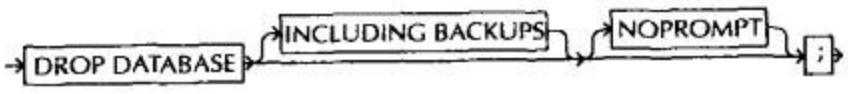


图 A-45 语法图

参考章节

第 16 章介绍了 drop database 命令的有关内容。

示例

```
drop database;
```

A.4.20　duplicate 命令

使用给定目标数据库的 RMAN 备份，执行 duplicate 命令既可以创建复制数据库(具有新的 DBID)，也可以创建备用数据库(具有相同的 DBID)，如图 A-46～图 A-51 所示。

语法图

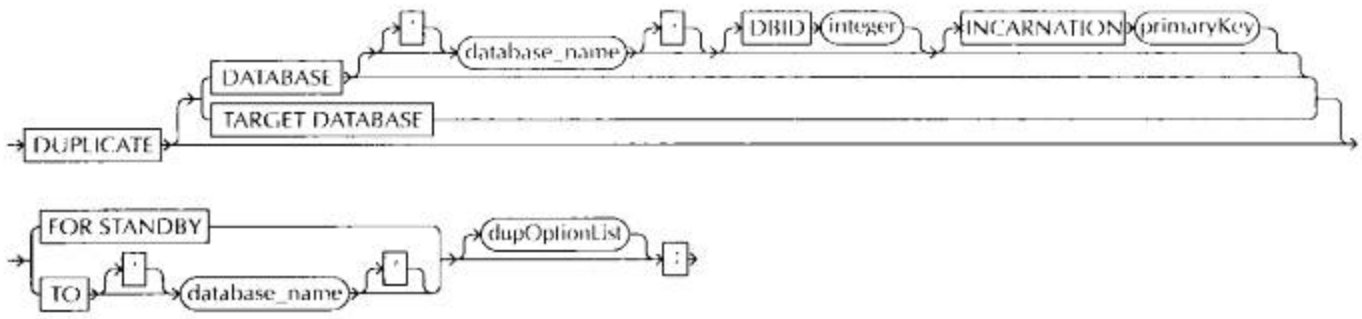


图 A-46　语法图

dupOptionList 参数语法图

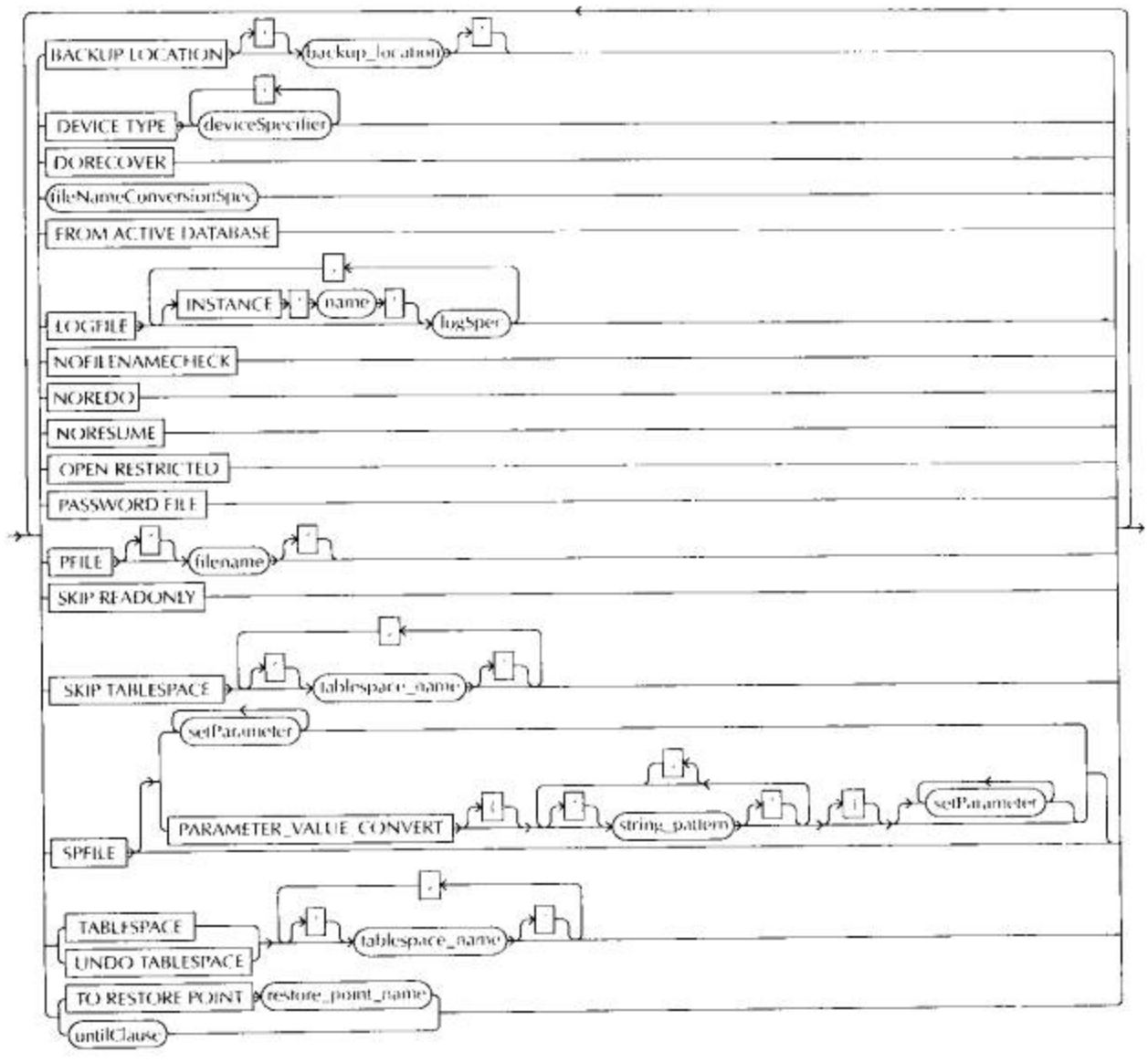


图 A-47　语法图

SetParameter 参数语法图

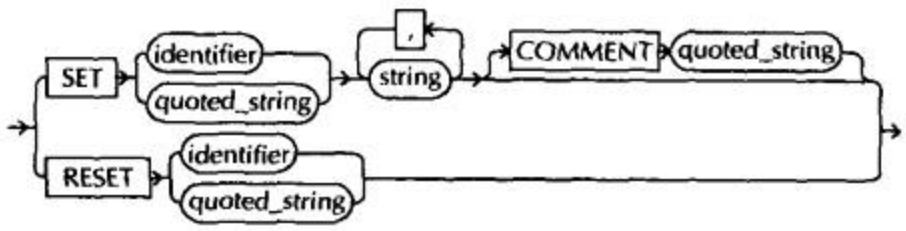


图 A-48 语法图

logSpec 参数语法图

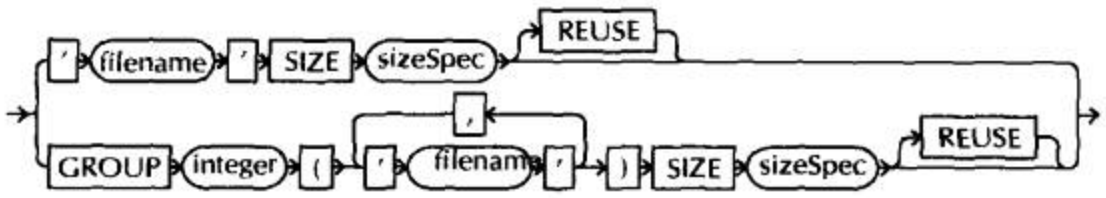


图 A-49 语法图

dupOptionList 参数语法图

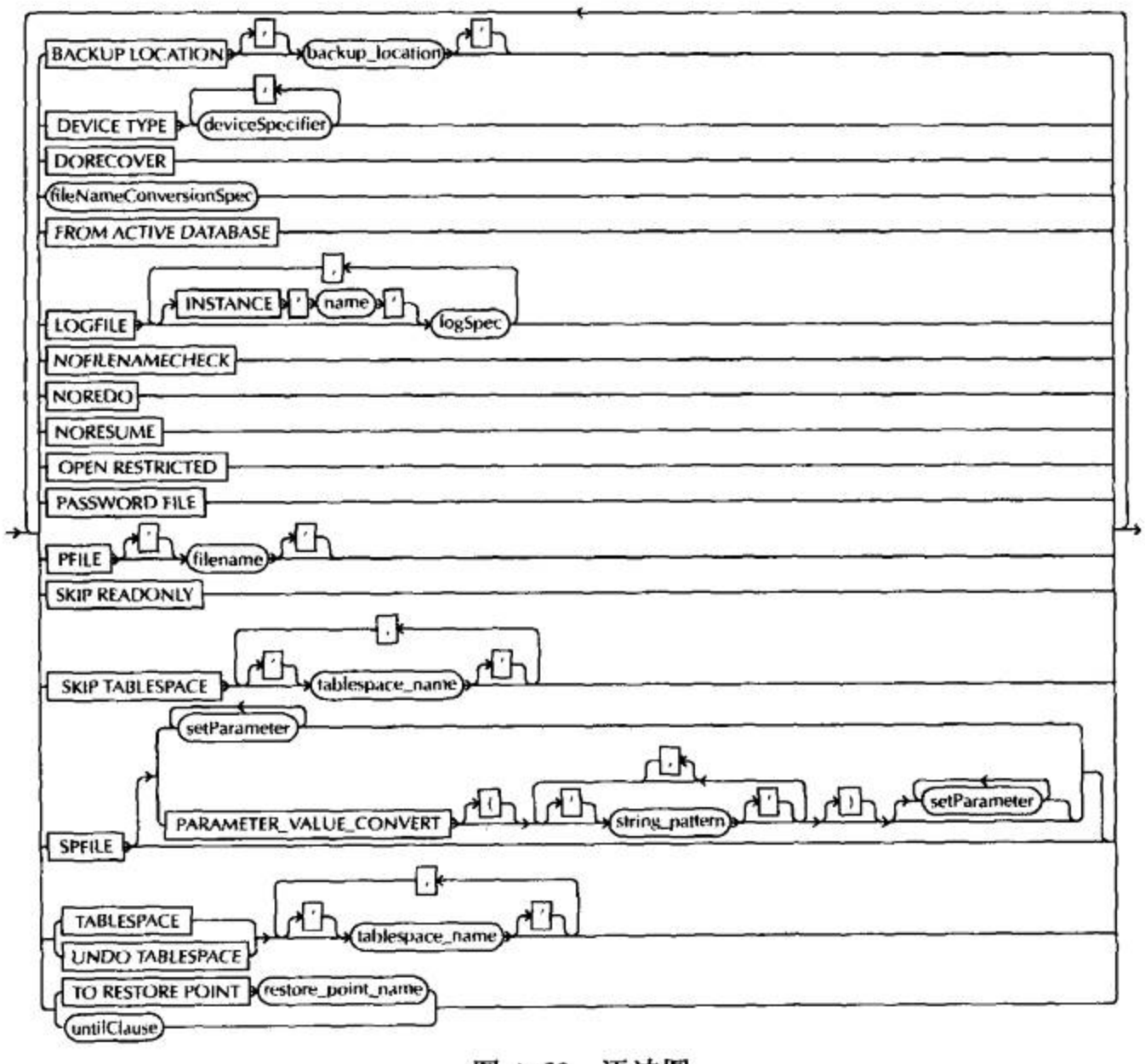


图 A-50 语法图

sizeSpec 参数语法图

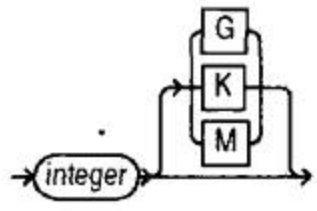


图 A-51　语法图

参考章节

第 19、20 章详细介绍了 duplicate 命令。

示例

```
duplicate target database to copydb
db_file_name_convert=
('/ora01/oracle/oradata/olddb/','/ora02/oracle/oradata/newdb/')
    until time 'sysdate- 5'
    skip tablespace oldusers, test
    pfile = ?/dbs/initcopydb.ora
    logfile
      group 1 ('/ora01/oraredo/newdb/redo01_1.f',
               '/ora02/oraredo/newdb/redo01_2.f') size 200k,
      group 2 ('/ora01/oraredo/newdb/redo02_1.f',
               '/ora02/oraredo/newdb/redo02_2.f') size 200k,
      group 3 ('/ora01/oraredo/newdb/redo03_1.f',
               '/ora02/oraredo/newdb/redo03_2.f') size 200k reuse;
```

A.4.21　execute script 命令

execute script 命令允许用户执行存储的 RMAN 脚本，如图 A-52 所示。

语法图

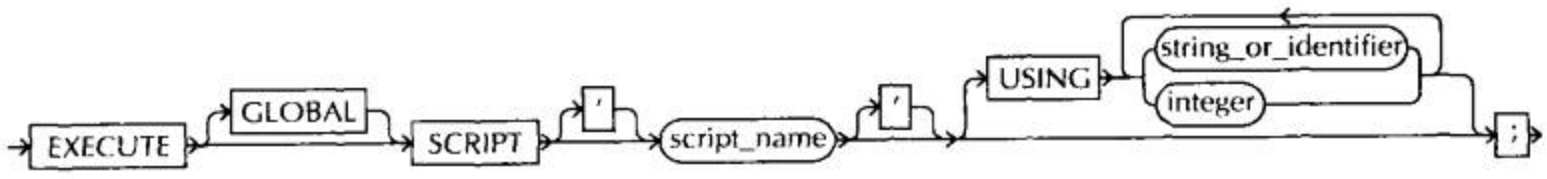


图 A-52　语法图

参考章节

第 10 章详细介绍了 execute script 命令。

示例

```
execute script my_script;
```

A.4.22 exit 命令

exit 命令可以使用户退出当前的 RMAN 会话操作，并且返回到 OS 提示符。

参考章节

第 3 章详细介绍了 exit 命令。

A.4.23 flashback database 命令

flashback database 命令允许 RMAN 管理 Oracle 闪回数据库操作。这就允许用户将数据库复位到之前的时间或 SCN，如图 A-53 所示。

语法图

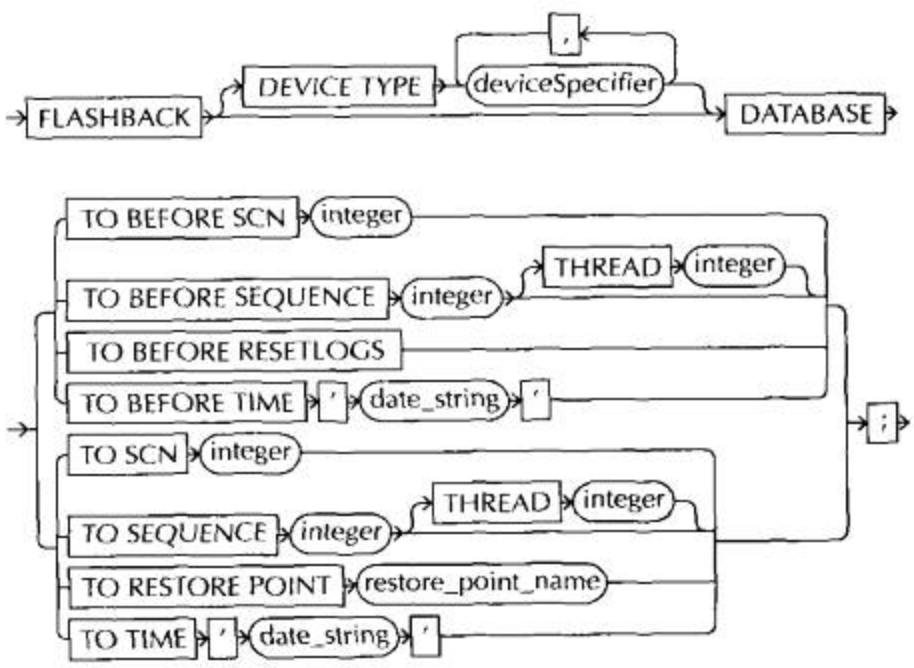


图 A-53 语法图

参考章节

第 15 章提供了 flashback 命令的详细信息。

示例

```
flashback database to scn 669605;
flashback database to timestamp (sysdate-2);
flashback database to timestamp
   to_timestamp('2006-01-03 15:00:00', 'yyyy-mm-dd hh24:mi:ss');
flashback database to restore point "gold_copy";
flashback database to before resetlogs;
```

A.4.24 grant 命令

grant 命令与 RMAN 恢复目录和称为虚拟私有目录的特性有关。该特性提供了一种由用户建立的对恢复目录中记录访问的限制策略。利用 grant 命令可以在配置了虚拟私有目录的情况下将查看恢复目录中记录的权限提供给用户，如图 A-54 所示。

语法图

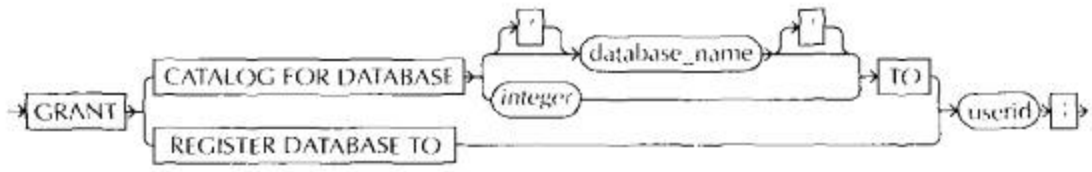


图 A-54　语法图

参考章节

第 9 章包含使用 grant 命令以及虚拟私有目录的有关信息。

示例

```
grant catalog for database mydb to operator1;
```

A.4.25　host 命令

host 命令用于从 RMAN 中调用 OS 命令行子命令解释程序。该命令也允许从 RMAN 提示符中运行特定的 OS 命令，如图 A-55 所示。

语法图

图 A-55　语法图

参考章节

第 16 章中的一些示例引用了 host 命令。

示例

```
host 'dir';
Host 'ls -al';
host
```

A.4.26 import catalog 命令

import catalog 命令从恢复目录模式中将元数据导出并导入到不同的目录模式中，如图 A-56 所示。

语法图

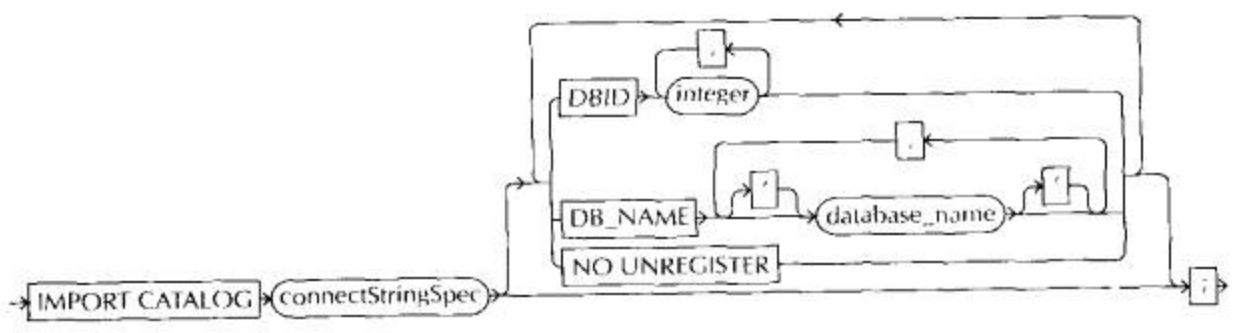


图 A-56　语法图

参考章节

第 10 章中介绍了该命令的示例。

示例

```
import catalog cat_owner@oldcatdb;
```

A.4.27 list 命令

list 命令提供 RMAN 备份的有关信息，例如何时进行备份、备份包含的文件，以及备份花费的时间，如图 A-57～图 A-61 所示。

语法图

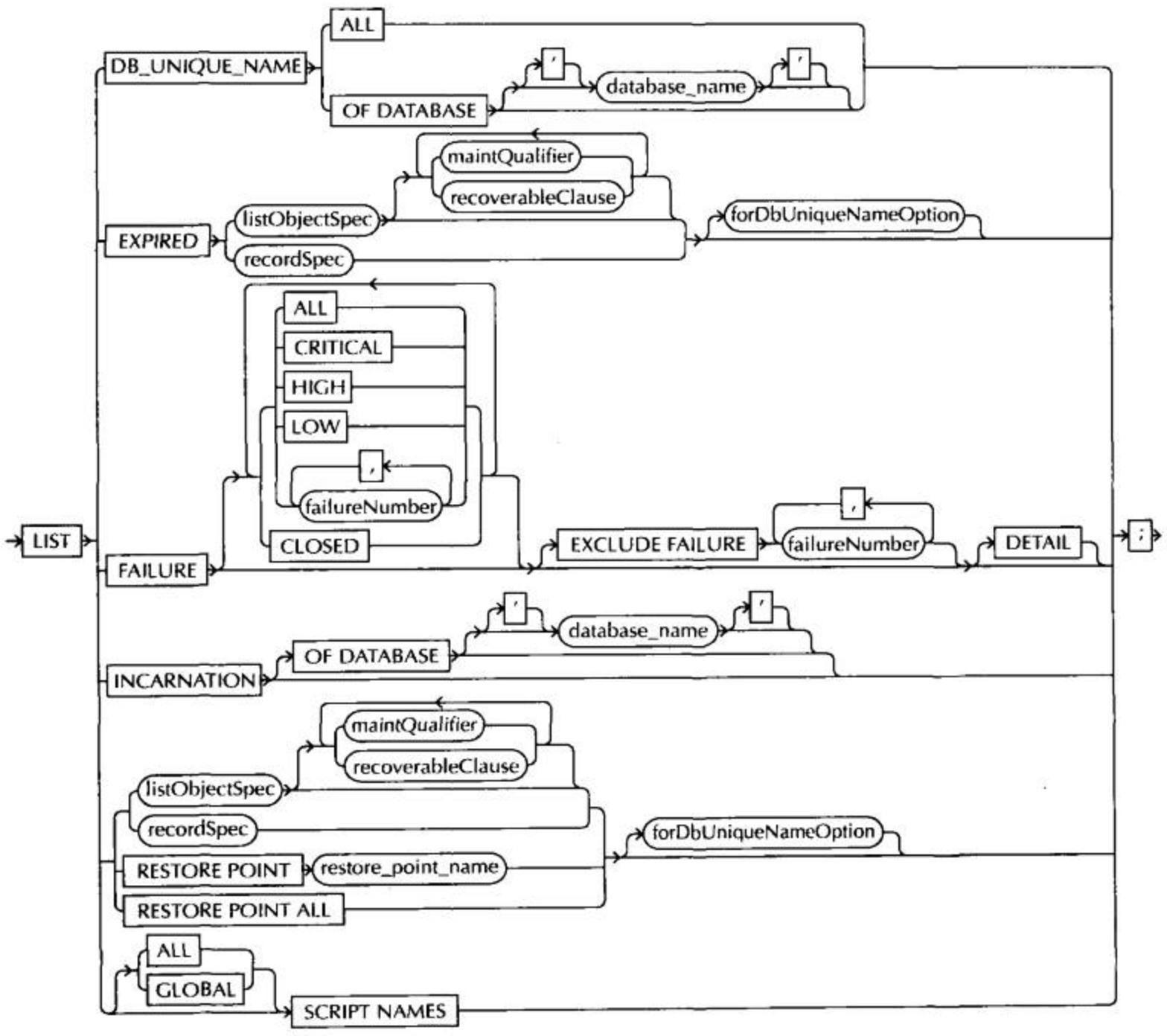


图 A-57　语法图

maintQualifier 参数语法图

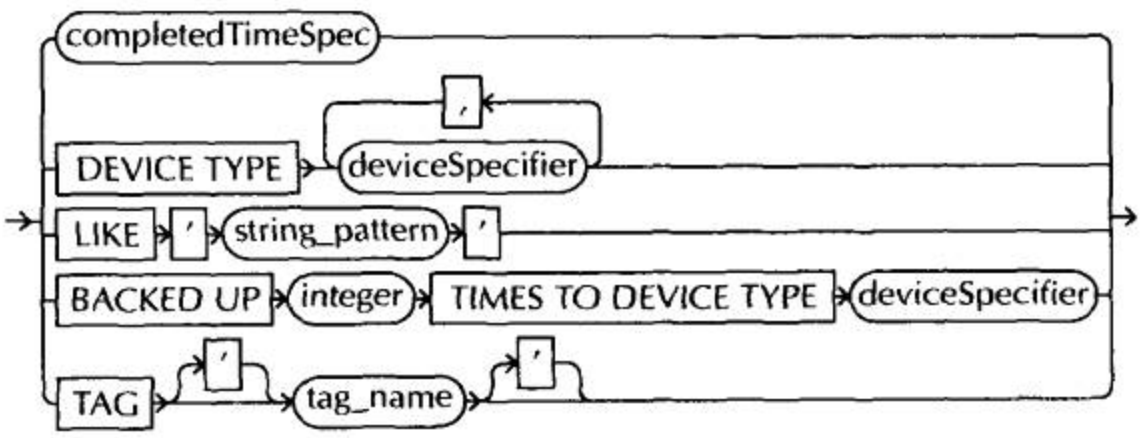


图 A-58　语法图

listObjectSpec 参数语法图

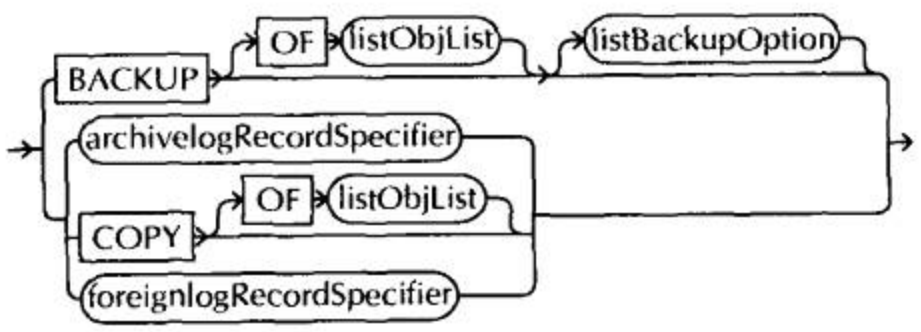


图 A-59　语法图

recoverableClause 参数语法图

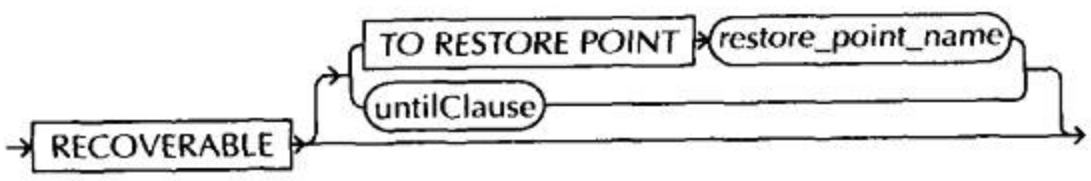


图 A-60　语法图

listBackupOption 参数语法图

图 A-61　语法图

参考章节

第 17 章介绍了与 RMAN 的 list 命令用法有关的内容。

示例

```
list backup of database;
list backup of database summary;
```

```
list backup of datafile;
list backup of archivelog all;
```

A.4.28 print script 命令

print script 命令用于显示存储的脚本，如图 A-62 所示。

语法图

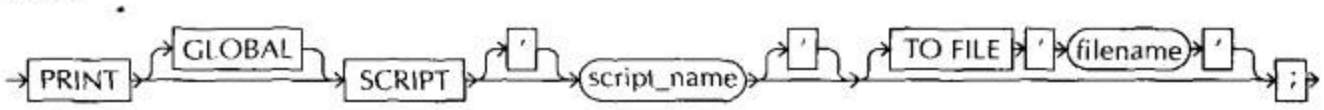


图 A-62 语法图

参考章节

第 10 章详细介绍了 print script 命令。

示例

```
print script booga_booga;
```

A.4.29 quit 命令

使用 quit 命令退出 RMAN 可执行程序。

参考章节

第 3 章详细介绍了 quit 命令。

A.4.30 recover 命令

recover 命令用于在还原后恢复数据库、表空间或数据文件。recover 命令在恢复过程中析取出所有必要的归档的重做日志，并且根据需要应用撤消操作，如图 A-63～图 A-69 所示。

语法图

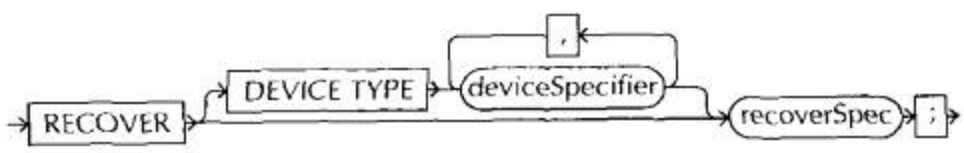


图 A-63 语法图

recoverSpec 参数语法图

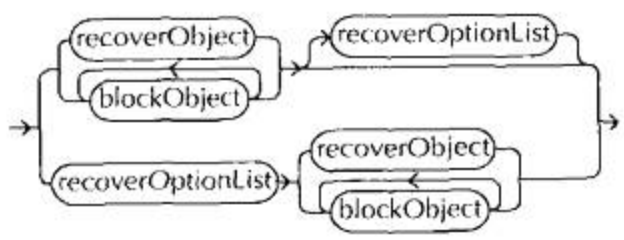


图 A-64 语法图

recoverObject 参数语法图

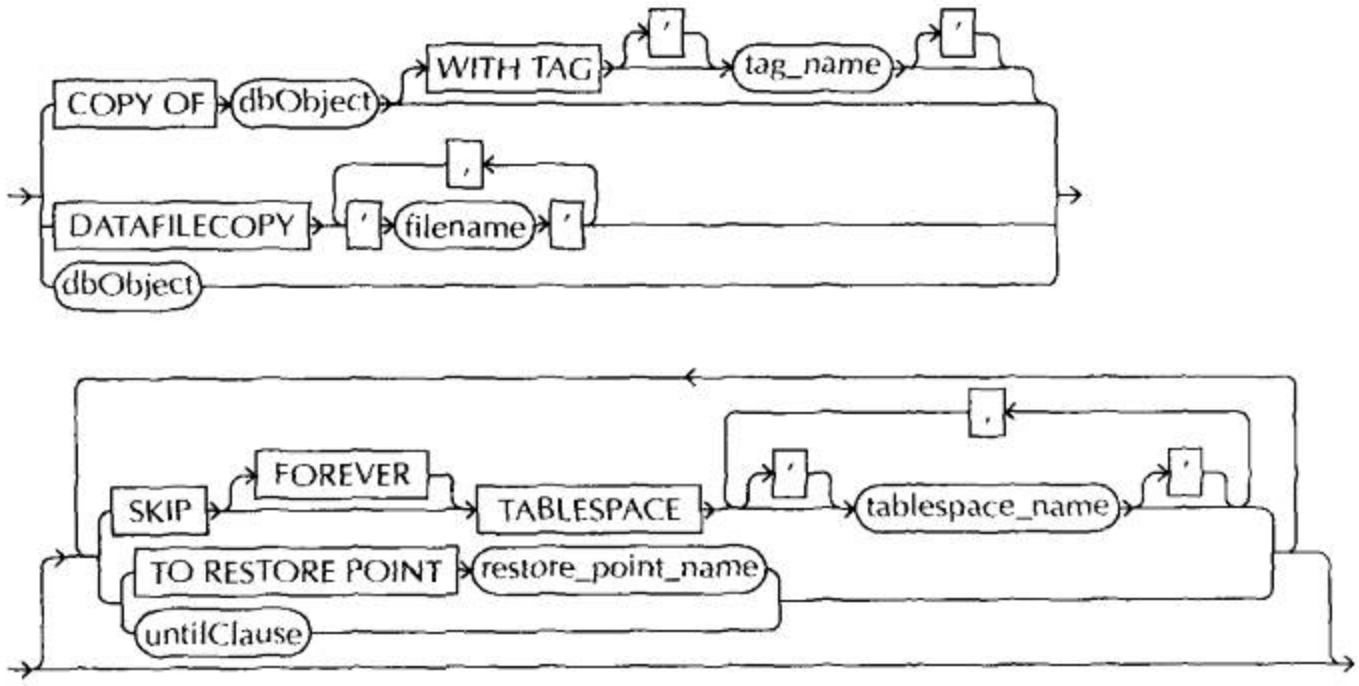


图 A-65　语法图

dbObject 参数语法图

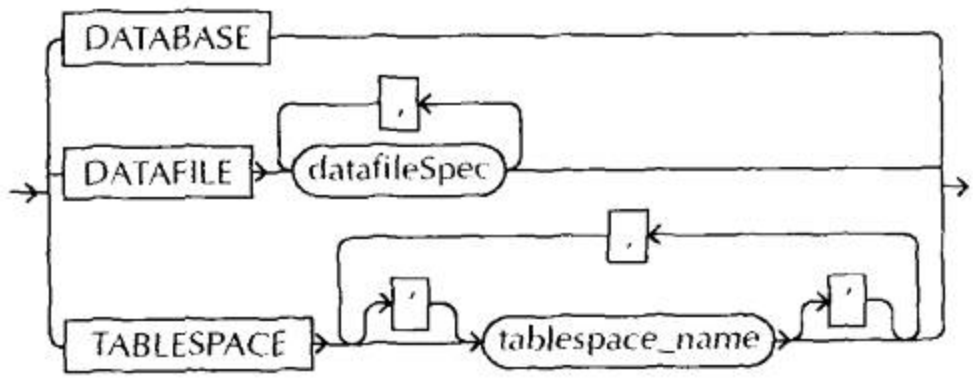


图 A-66　语法图

blockObject 参数语法图

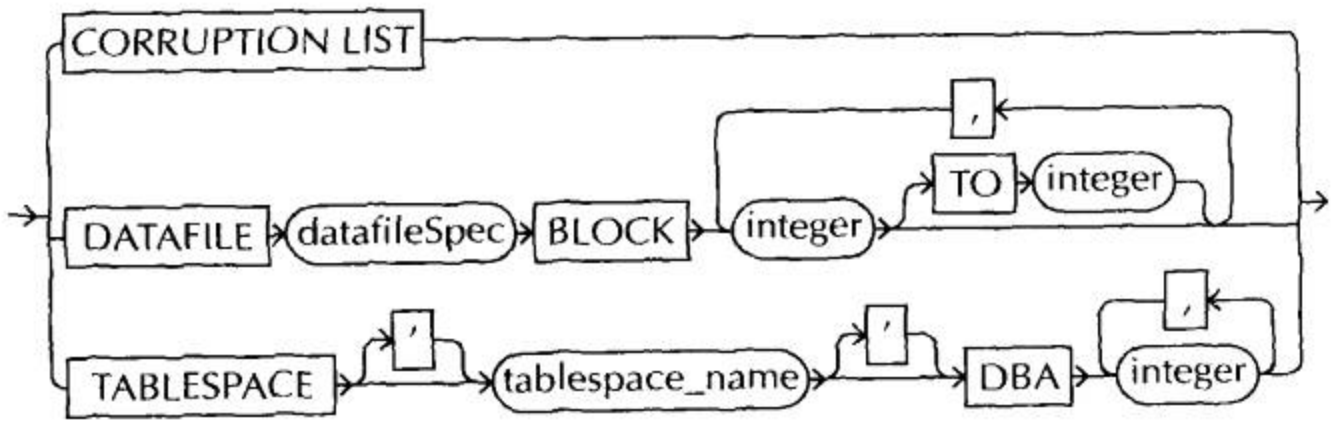


图 A-67　语法图

recoverOptionList 参数语法图

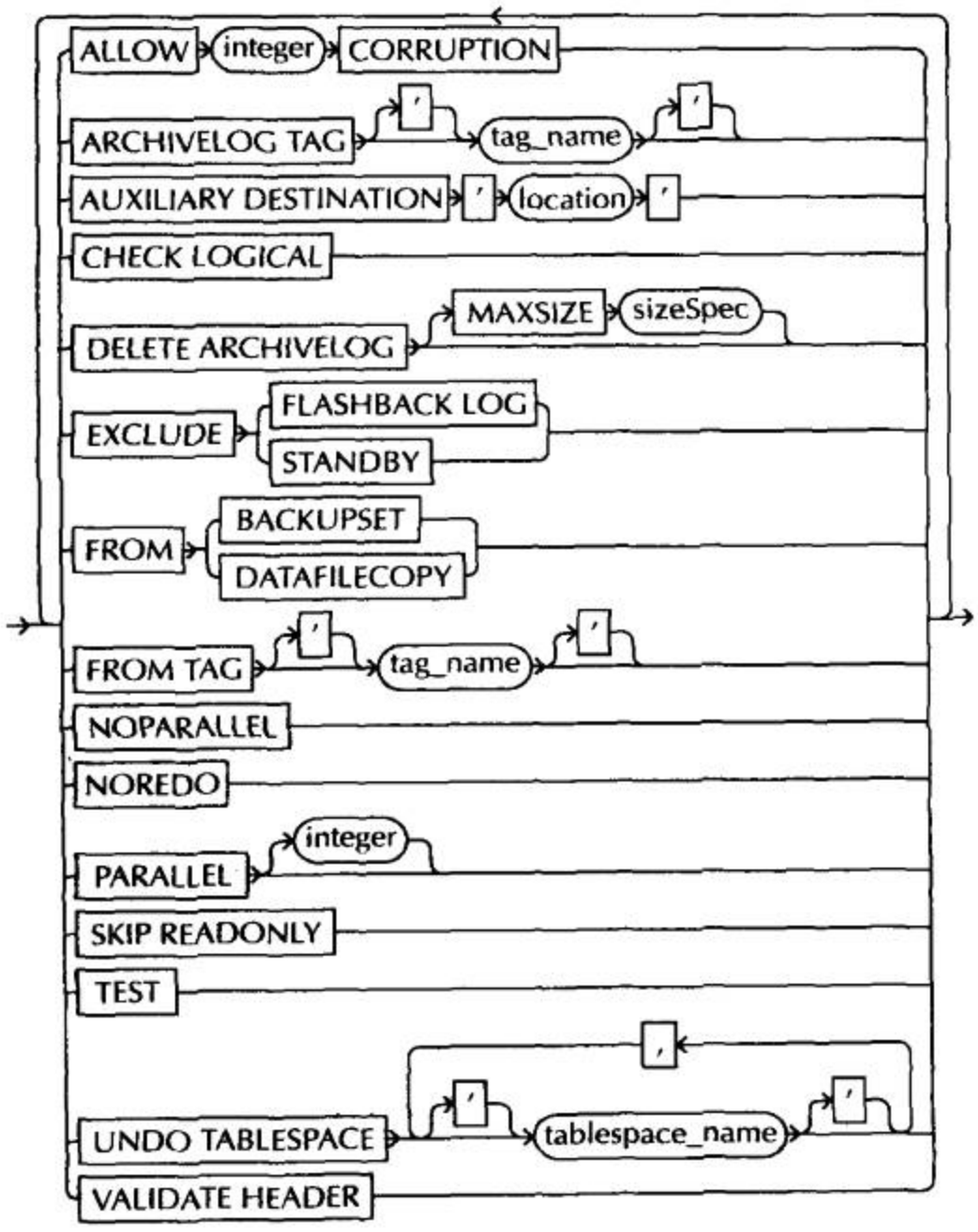


图 A-68　语法图

sizeSpec 参数语法图

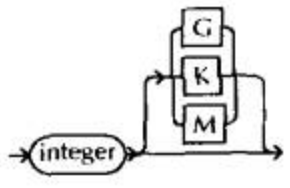


图 A-69　语法图

参考章节

第 12、14 章等章节详细介绍了 recover 命令。

示例

```
recover database;
recover tablespace users;
recover datafile 1;
```

A.4.31　register database 命令

register database 命令用于在已有的恢复目录中注册目标数据库，如图 A-70 所示。

语法图

图 A-70　语法图

参考章节

第 10 章介绍了 register database 命令。

示例

```
register database;
```

A.4.32 release channel 命令

release channel 命令用于释放使用 allocate channel 命令分配的信道(常规信道和维护信道)。通常，RMAN 会自动释放信道，因此在实际中很少用到该命令，如图 A-71～图 A-72 所示。

语法图

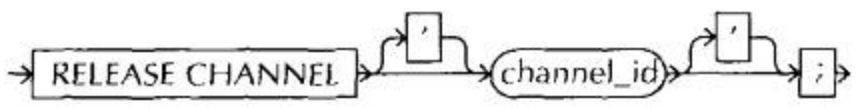


图 A-71　语法图

releaseForMaint 参数语法图

图 A-72　语法图

示例

```
release channel c1;
```

A.4.33　repair failure 命令

repair failure 命令用于修改由数据恢复顾问(DRA)标识的错误，如图 A-73 所示。

语法图

图 A-73　语法图

参考章节

第 12 章包含 repair failure 命令的有关信息。

示例

```
repair failure;
```

A.4.34　replace script 命令

replace script 命令用于替换存储在恢复目录中的命令脚本，如图 A-74 所示。

语法图

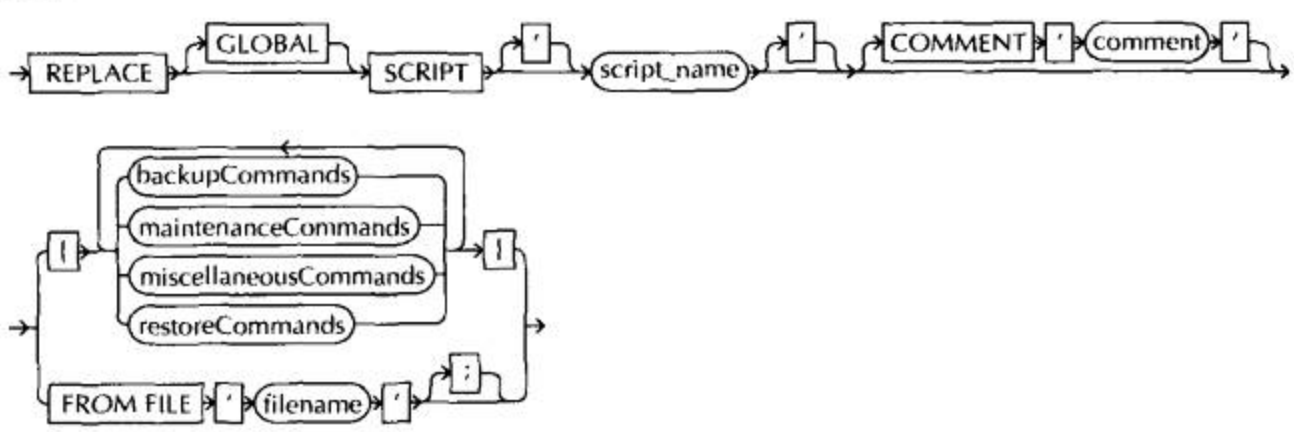


图 A-74　语法图

参考章节

第 10 章详细介绍了该命令。

示例

```
replace script backup_full
    { backup device type sbt database plus archivelog; }
```

A.4.35　report 命令

report 命令用于报告基于恢复目录或数据库控制文件中的元数据得出的功能。report 命令的输出能够用于确定数据库及其组件部分的恢复状态，如图 A-75～图 A-78 所示。

语法图

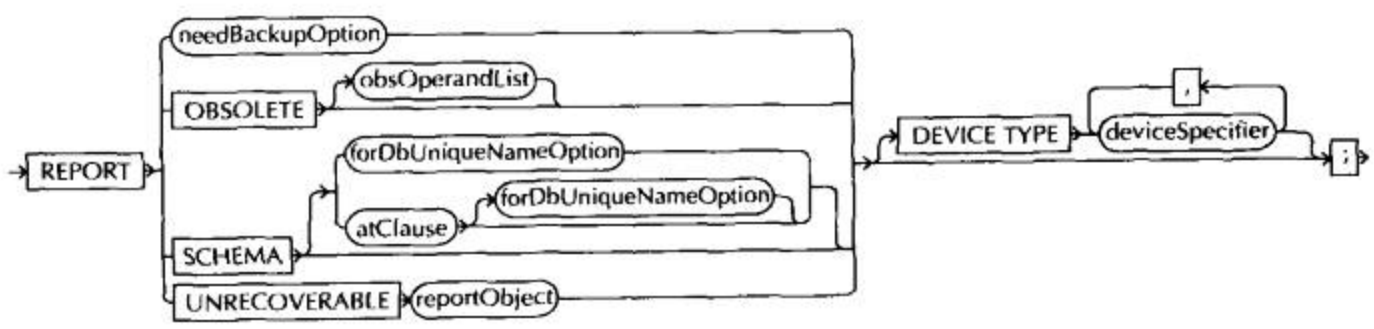


图 A-75　语法图

needBackupOption 参数语法图

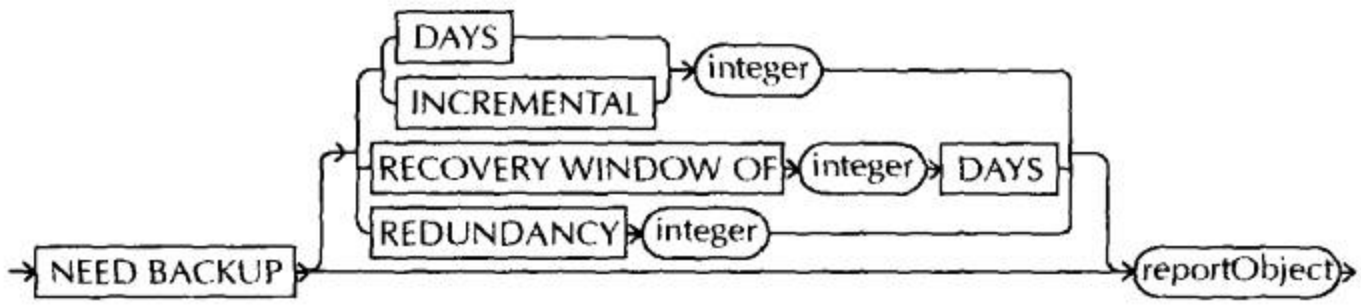


图 A-76　语法图

reportObject 参数语法图

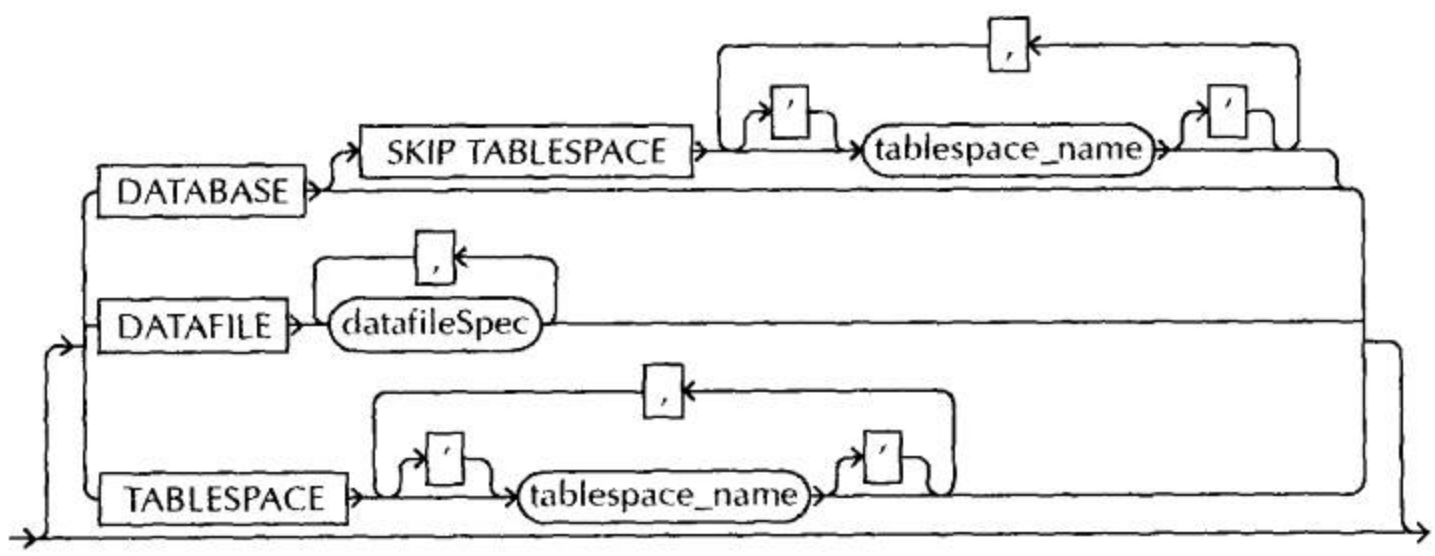


图 A-77　语法图

atClause 参数语法图

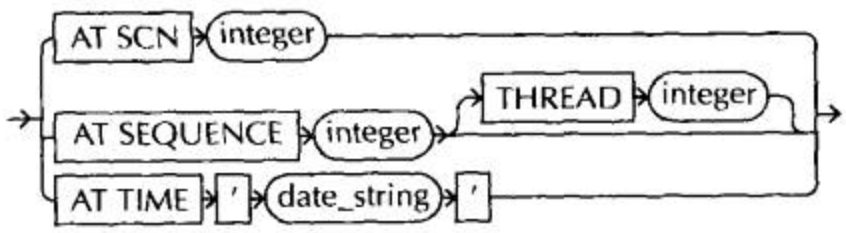


图 A-78　语法图

参考章节

第 17 章详细介绍了 report 命令。

示例

```
report need backup days=3;
report unrecoverable;
report obsolete;
```

A.4.36　reset database 命令

在使用 alter database open resetlogs 命令创建新的数据库对应物之后，将使用 reset database 命令。reset database 命令也可将数据库复位到前面的数据库对应物以用于恢复，

如图 A-79 所示。

语法图

图 A-79　语法图

参考章节

第 14 章详细介绍了 reset database 命令。

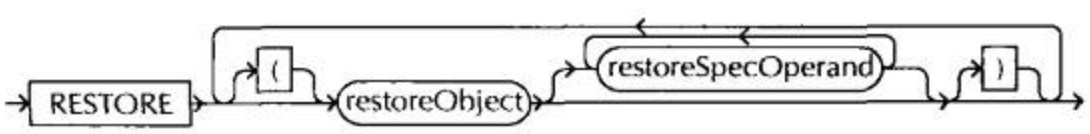
示例

```
reset database to incarnation 3;
reset database;
```

A.4.37　restore 命令

restore 命令用于从备份介质中还原数据库数据文件、数据文件副本、归档重做日志、控制文件以及服务器参数文件，为恢复操作做准备，如图 A-80～图 A-83 所示。

语法图



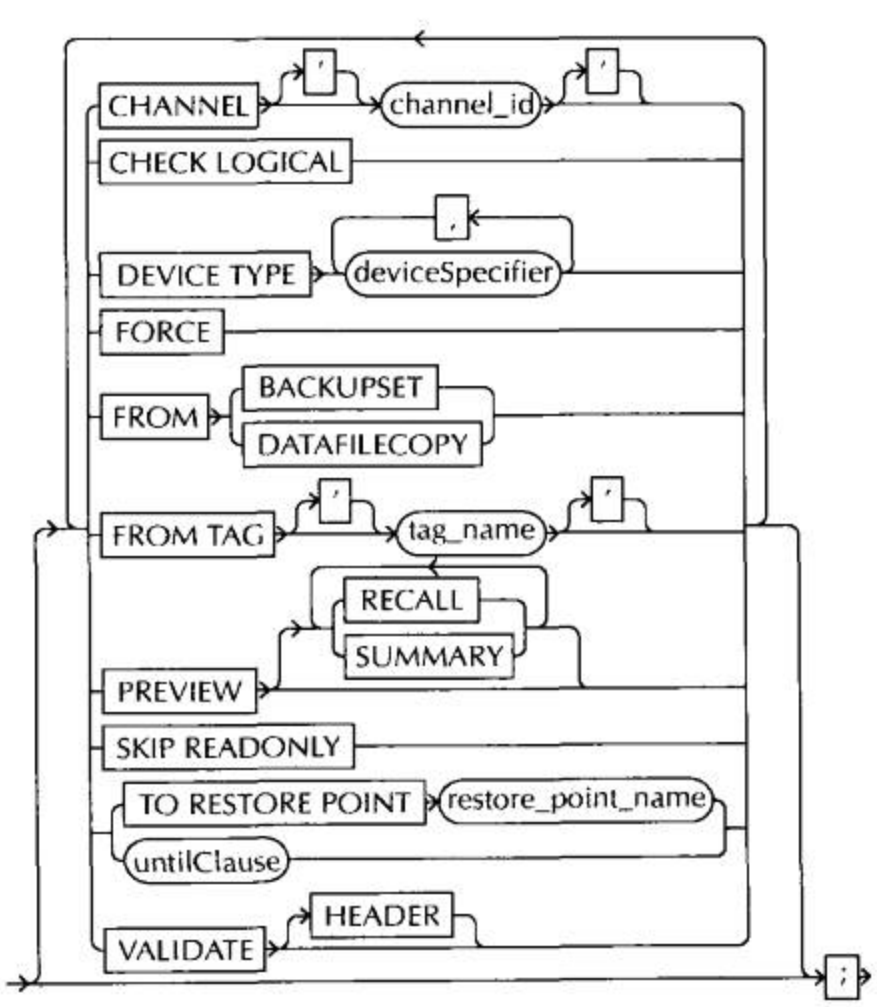


图 A-80　语法图

restoreObject 参数语法图

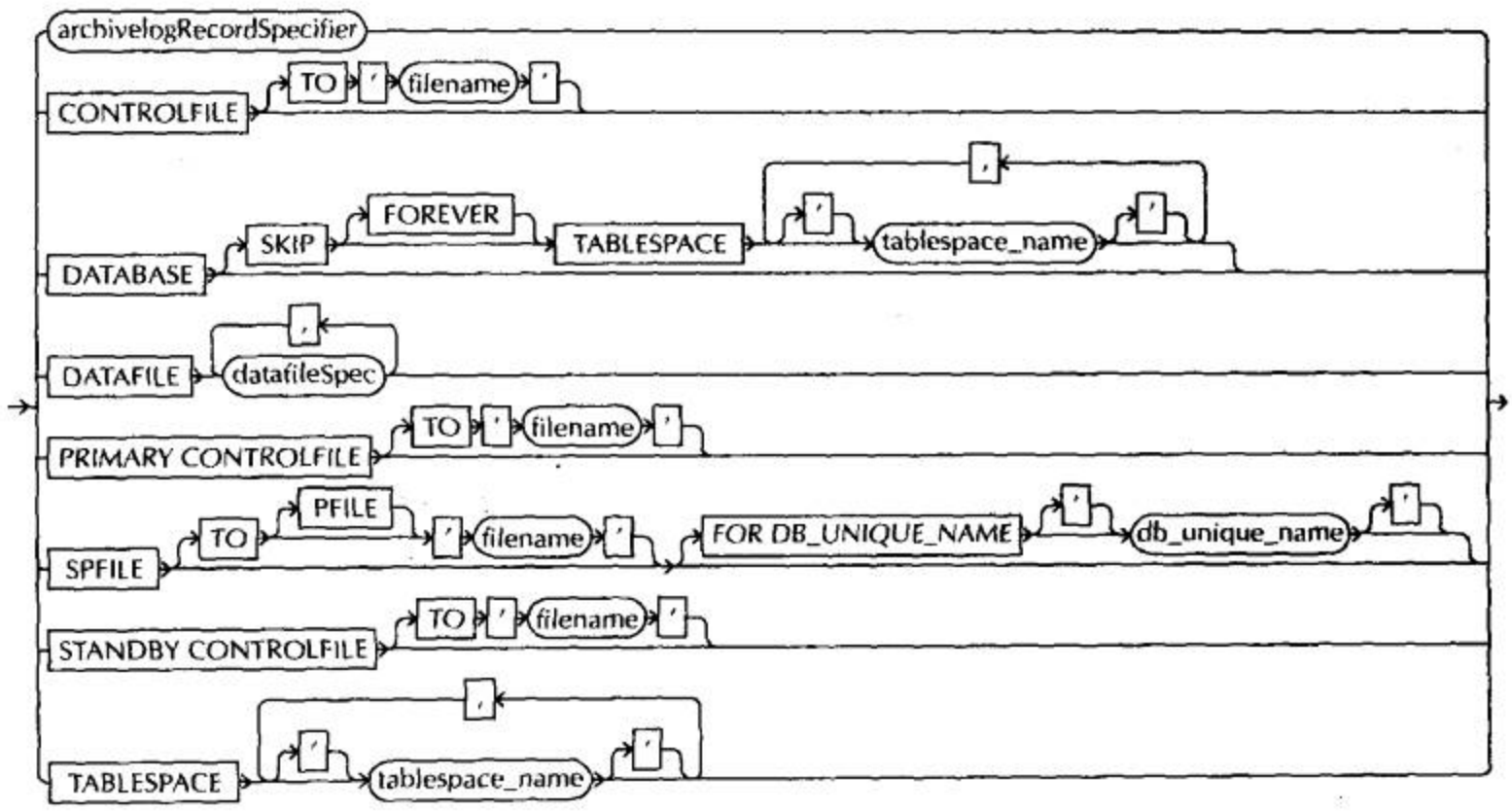


图 A-81　语法图

restoreSpecOperand 参数语法图

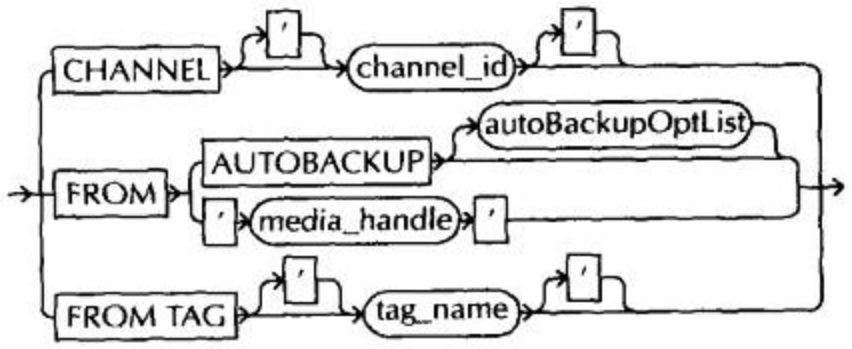


图 A-82　语法图

autoBackupOptList 参数语法图

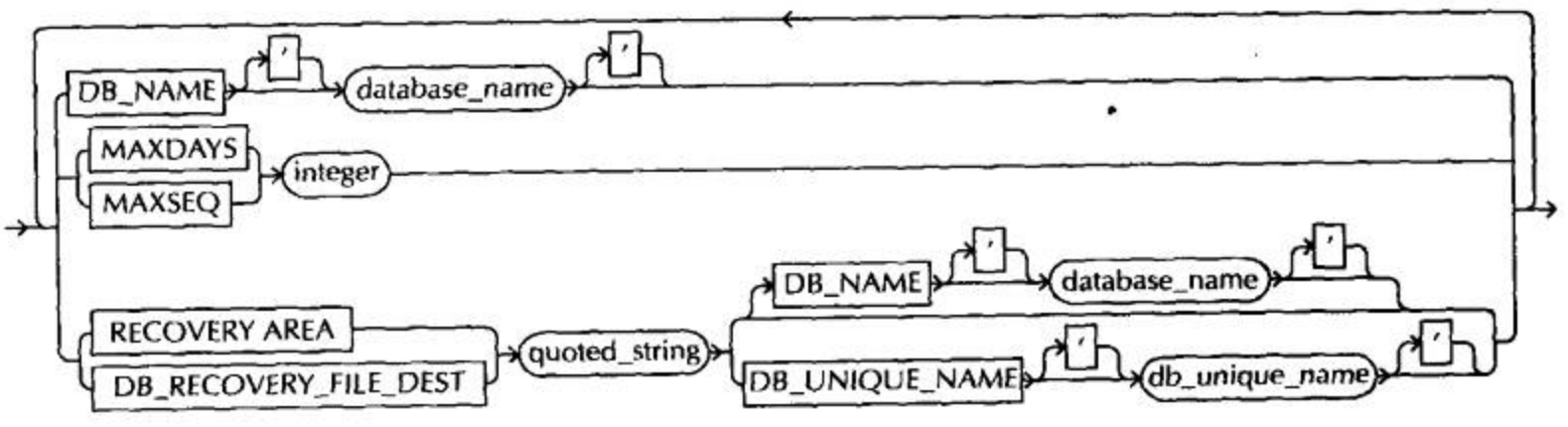


图 A-83　语法图

参考章节

第 12、14 章详细介绍了 restore 命令。

示例

```
restore database;
restore tablespace user_data, user_index;
restore datafile 1,2,3,4,5;

CONNECT TARGET /
STARTUP NOMOUNT;
SET DBID 660055374;
RESTORE CONTROLFILE FROM TAG 'monday_cf_backup';
ALTER DATABASE MOUNT;

RMAN> RUN
{
  SET ARCHIVELOG DESTINATION TO '/oratmp/arch_restore';
  RESTORE ARCHIVELOG ALL;
}
```

A.4.38　resync catalog 命令

resync catalog 命令用于再同步恢复目录与数据库控制文件，如图 A-84 所示。

语法图

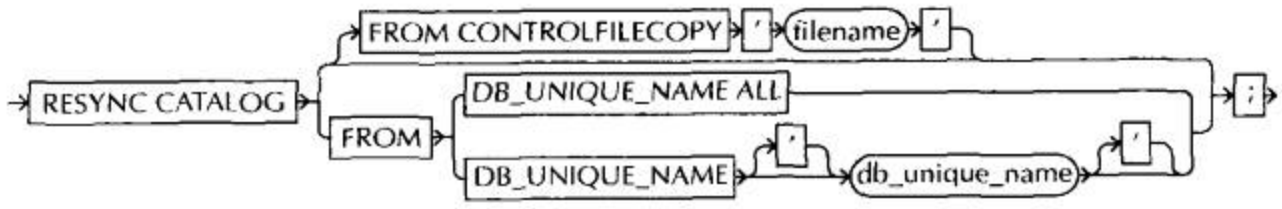


图 A-84　语法图

参考章节

第 10 章介绍了 resync 命令的有关信息。

示例

```
resync catalog;
```

A.4.39　revoke 命令

revoke 命令与 RMAN 恢复目录和称为虚拟私有目录的特性有关。该特性提供了一种由用户建立的对恢复目录中记录访问的限制策略。前面提到，利用 grant 命令可以在配置了虚拟私有目录的情况下将查看恢复目录中记录的权限提供给用户。而用 revoke 命令可以将提供给用户的查看权限撤消，如图 A-85 所示。

语法图

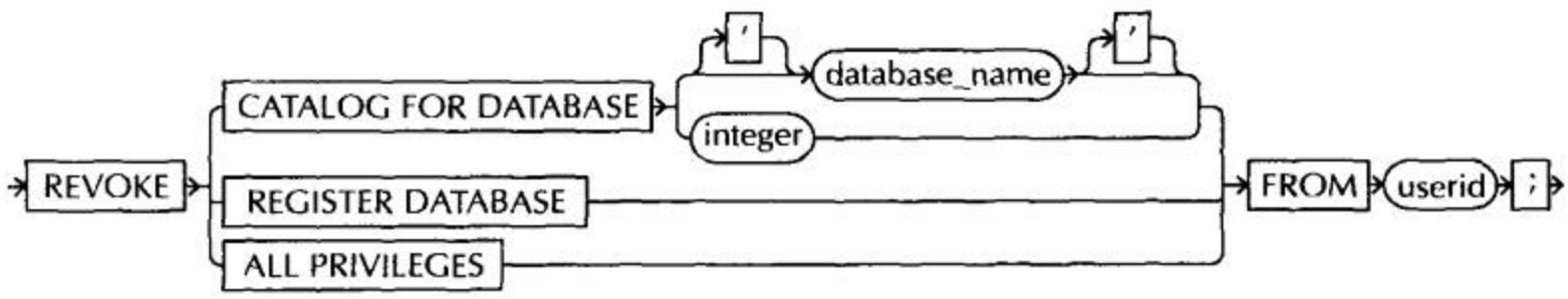


图 A-85　语法图

参考章节

第 10 章包含 revoke 命令和使用虚拟私有目录的相关信息。

示例

```
revoke catalog for database mydb from operator1;
revoke all privileges from operator2;
```

A.4.40　run 命令

run 命令用于编译和执行在 run 代码块花括号范围内的一条或多条 RMAN 命令。在实际运用中，run 命令后面是一个表示开始的花括号，随后是一条或多条 RMAN 命令。一旦列出了最后的 RMAN 命令，此时按下 ENTER 键即可使表示结束的花括号执行整组命令，如图 A-86～图 A-90 所示。

语法图

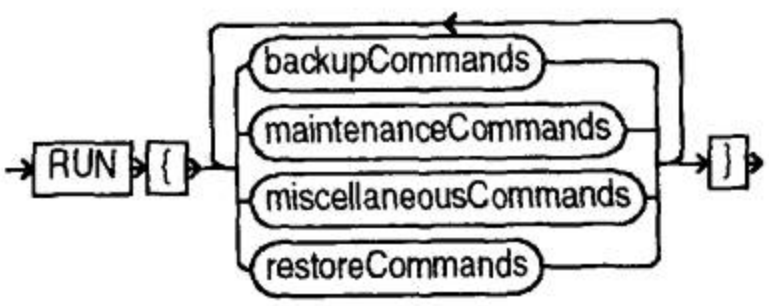


图 A-86　语法图

backupCommands 参数语法图

图 A-87　语法图

maintenanceCommands 参数语法图

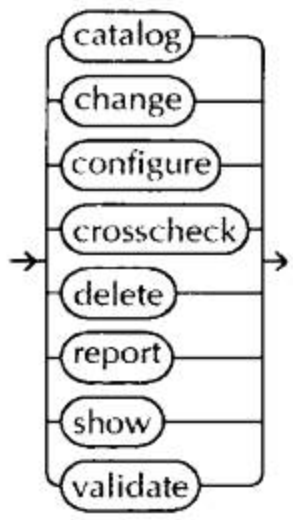


图 A-88　语法图

miscellaneousCommands 参数语法图

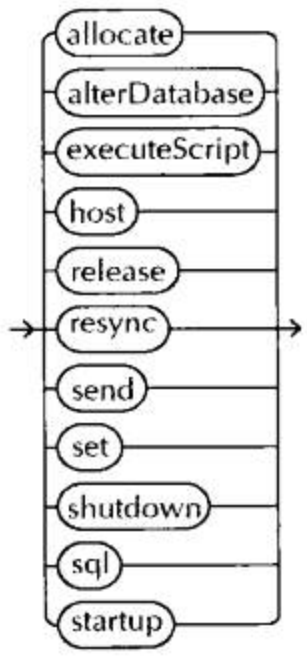


图 A-89　语法图

restoreCommands 参数语法图

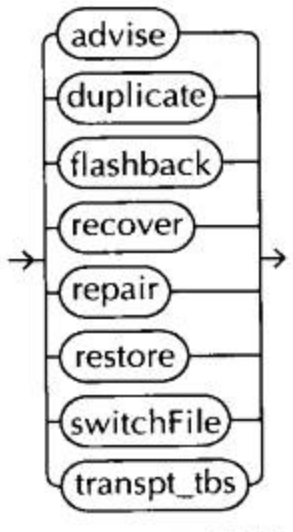


图 A-90　语法图

参考章节

本书的许多章节都示例说明了 run 命令，例如第 11 章。

示例

```
RMAN> RUN
{
  set archivelog destination to '/oratmp/arch_restore';
  restore archivelog all;
}
```

A.4.41　send 命令

send 命令用于向一条或多条信道发送供应商指定的字符串。由于这些字符串是由供应商指定的，因此应当记录在供应商所提供的介质管理层(MML)文档中，如图 A-91 所示。

语法图

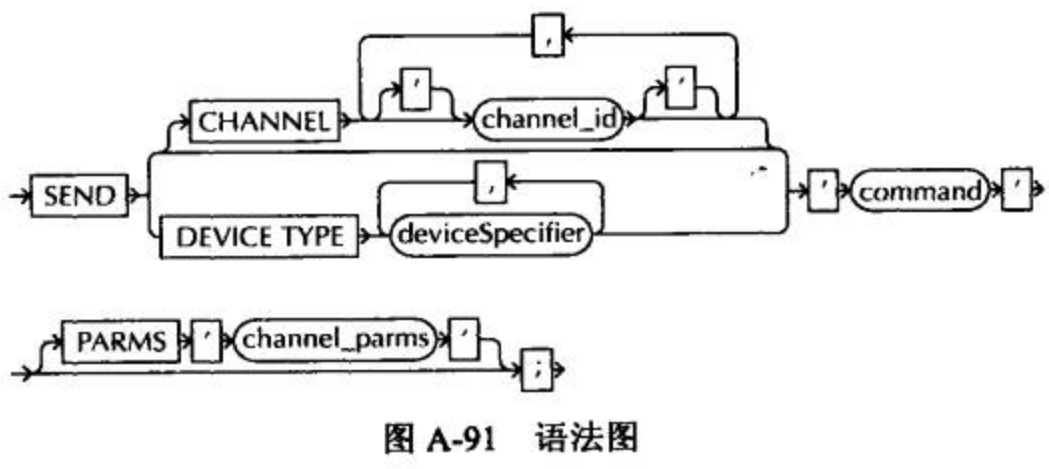


图 A-91　语法图

参考章节

第 4 章到第 9 章详细介绍了 MML 层和有效供应商字符串的信息。

示例

```
send 'env=serv_bck01';
```

A.4.42　set 命令

set 命令为当前的 RMAN 会话配置 RMAN 设置。它用于配置非永久的设置，或者临时更改给指定会话的永久设置。set 命令提供两组不同的形式。第一组参数 set_run_option 只在 run 代码块的范围内有效；第二组参数 set_rman_option 只在 RMAN 提示符中有效，如图 A-92～图 A-95 所示。

语法图

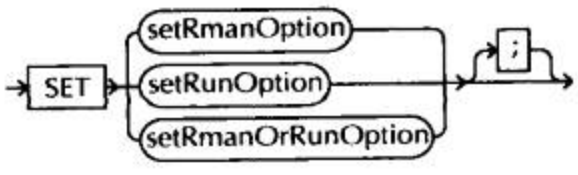


图 A-92　语法图

setRmanOption 参数语法图

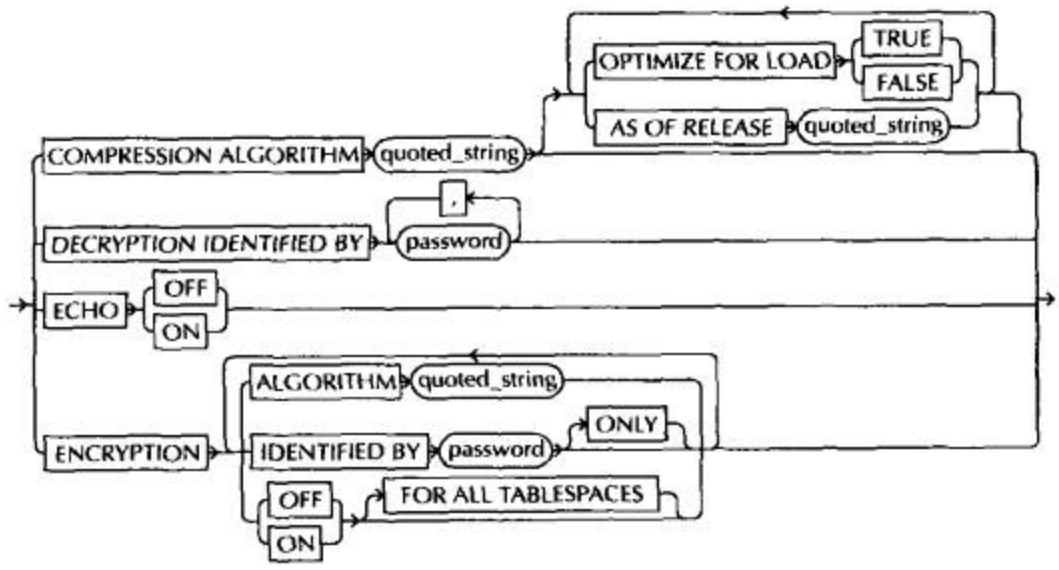


图 A-93 语法图

setRunOption 参数语法图

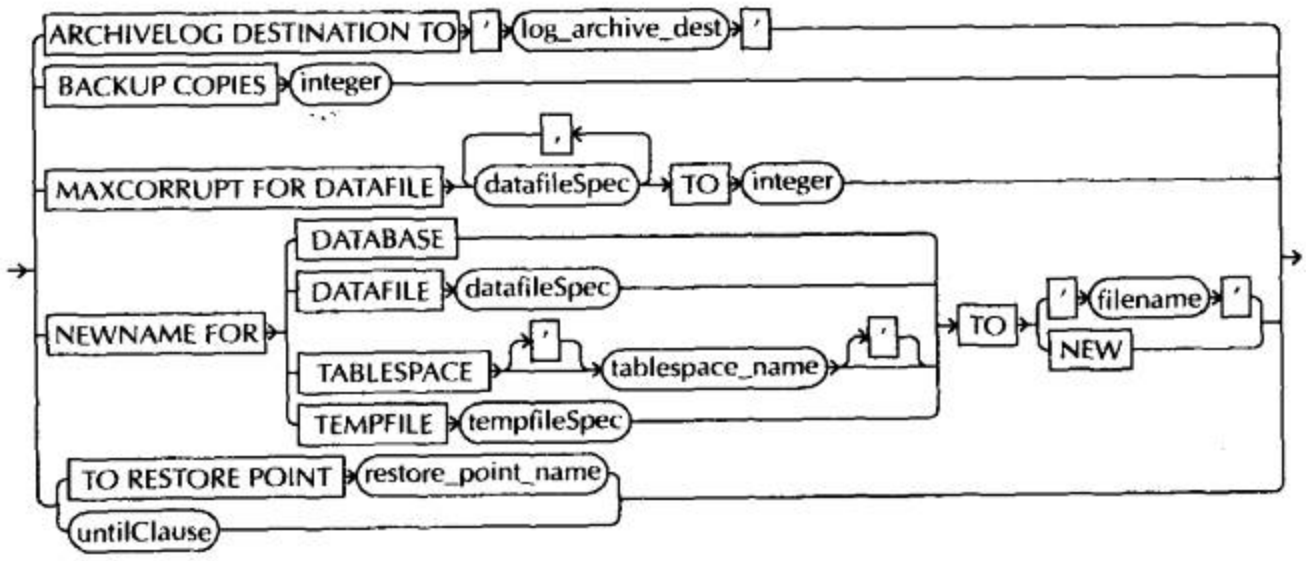


图 A-94 语法图

setRmanOrRunOption 参数语法图

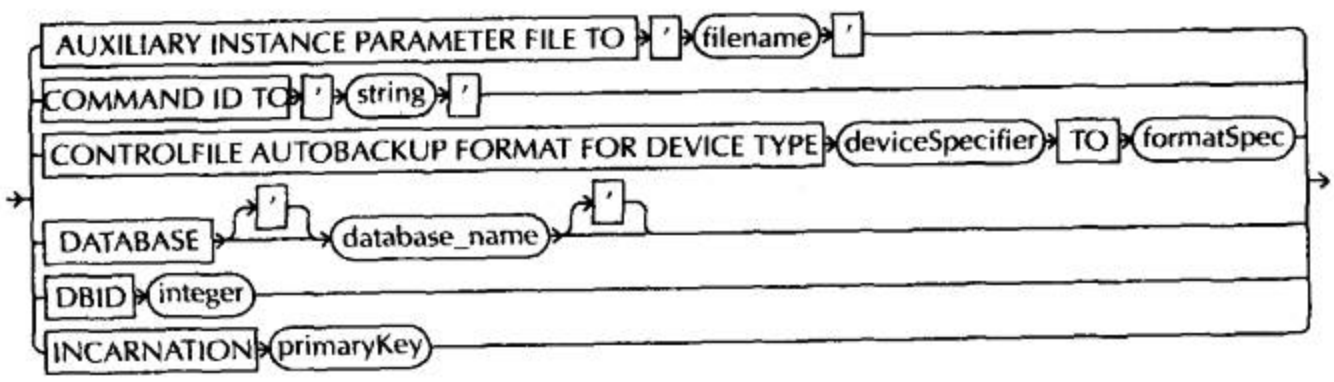


图 A-95 语法图

参考章节

第 12 章详细介绍了 set 命令。

示例

```
set dbid 676549873;
set until time '1/10/2006 13:45:00';
set controlfile autobackup format for device type disk
to '/u01/oraback/cntrlf_%f.bak';
```

A.4.43　show 命令

show 命令用于显示 RMAN 连接的数据库当前永久配置的设置。这些设置来自于数据库的控制文件，如图 A-96～图 A-97 所示。

语法图

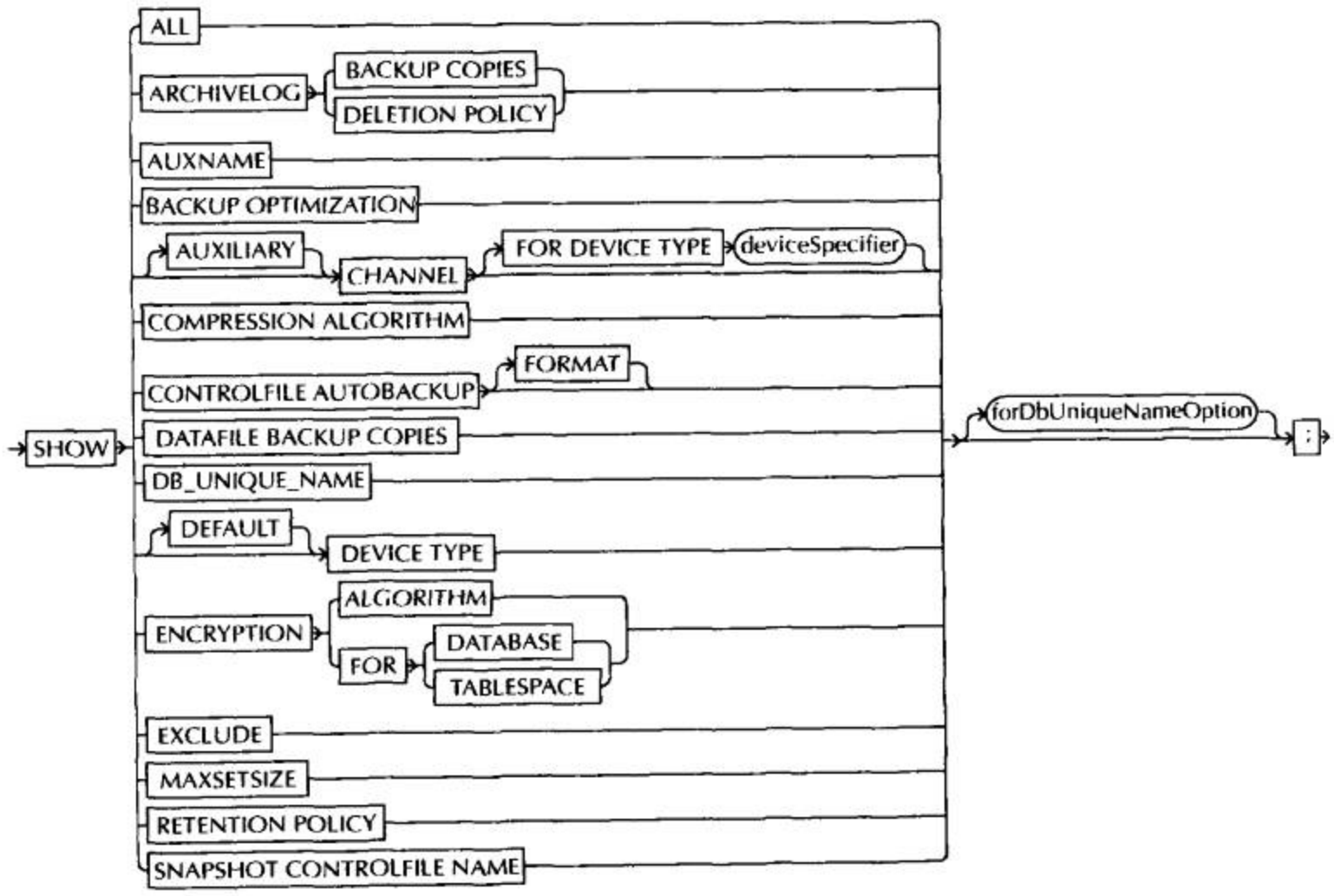


图 A-96　语法图

forDbUniqueNameOption 参数语法图

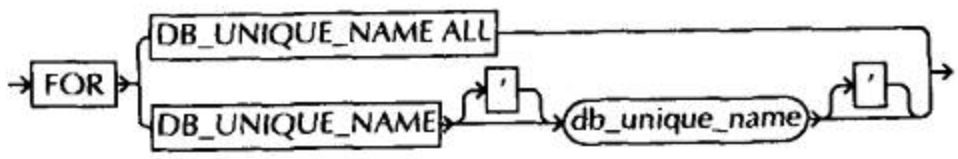


图 A-97　语法图

参考章节

第 3 章示例说明了 show 命令。

示例

```
show all;
show retention policy;
```

A.4.44 shutdown 命令

shutdown 命令用于关闭目标数据库，等同于 SQL *Plus 的 shutdown 命令，如图 A-98 所示。

语法图

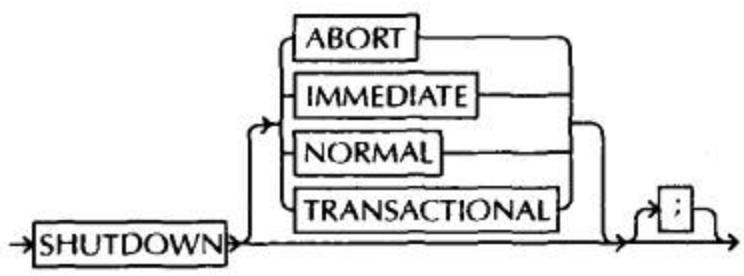


图 A-98 语法图

参考章节

本书的许多章节都示例说明了该命令。

示例

```
shutdown abort;
shutdown immediate;
```

A.4.45 spool 命令

spool 命令用于将 RMAN 操作的输出缓存到一个操作系统文件，如图 A-99 所示。

语法图

图 A-99 语法图

示例

```
spool log to 'rmanoutput.log'
```

A.4.46 SQL 命令

SQL 命令用于在 RMAN 提示符下执行 SQL 语句，在单引号中放置一条 SQL 语句，如图 A-100 所示。

语法图

图 A-100　语法图

参考章节

本书的许多章节都示例说明了该命令。

示例

```
sql 'alter tablespace mytbs offline immediate';
```

A.4.47　startup 命令

startup 命令用于启动目标数据库，等同于 SQL *Plus 中的 startup 命令，如图 A-101 所示。

语法图

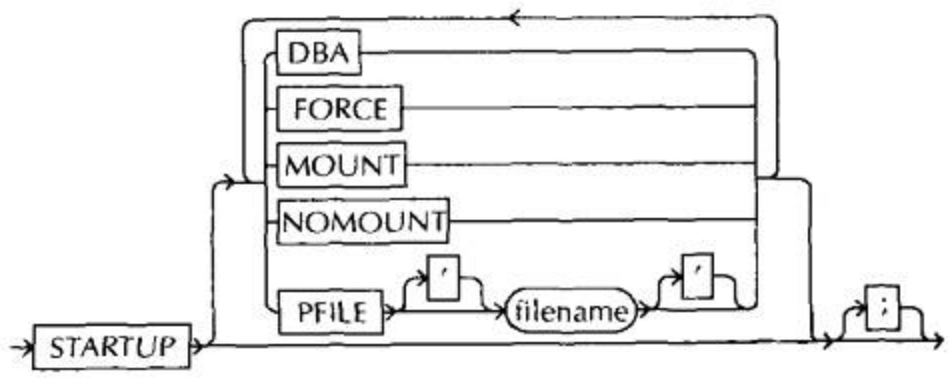


图 A-101　语法图

参考章节

本书的许多章节都示例说明了该命令。

示例

```
startup nomount;
startup;
```

A.4.48　switch 命令

switch 命令使 RMAN 数据文件副本变为当前的数据库数据文件。如果这个数据库数据文件需要被还原到不同于原始位置的位置中，并且与 RMAN 的 set newname 命令相关联，则通常需要使用 switch 命令。当应用于数据库数据文件时，该命令等同于 SQL 的 alter database rename file 语句，如图 A-102～图 A-103 所示。

语法图

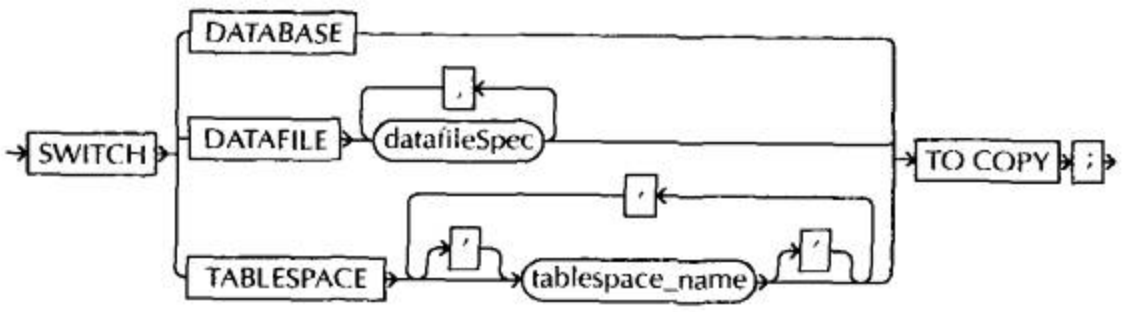


图 A-102　语法图

switchFile 参数语法图

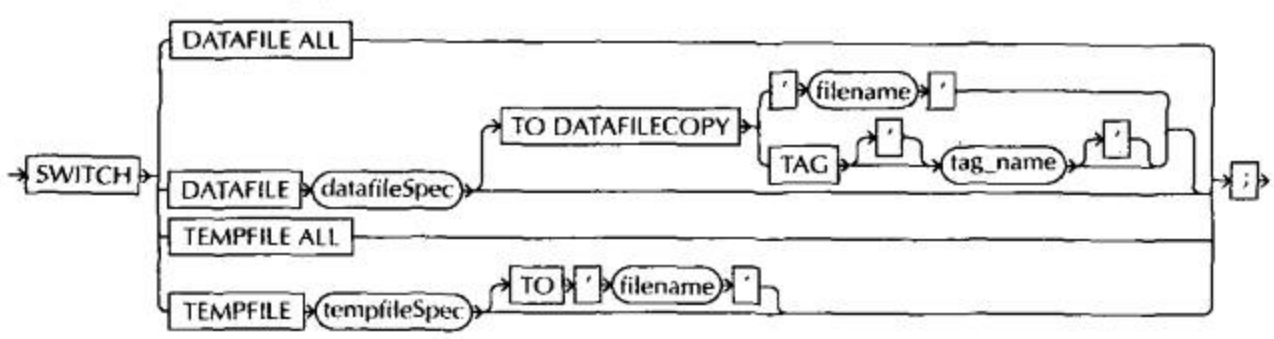


图 A-103　语法图

参考章节

第 14 章详细介绍了该命令的用法。

示例

```
run
{
  set newname for datafile '/disk7/oracle/tbs11.f'
    to '/disk9/oracle/tbs11.f';
  restore tablespace tbs_1;
  switch datafile all;
  recover tablespace tbs_1;
  sql "alter tablespace tbs_1 online";
}
```

A.4.49　transport tablespace 命令

transport tablespace 命令用于从 RMAN 备份中创建可传递的表空间集，而不是使用源数据库的实际数据文件。也可以使用 transport tablespace 命令创建可传递的表空间集，可以将该表空间集恢复到特定的时间点，如图 A-104～图 A-105 所示。

语法图

图 A-104　语法图

Transp_tbs_optlist 参数语法图

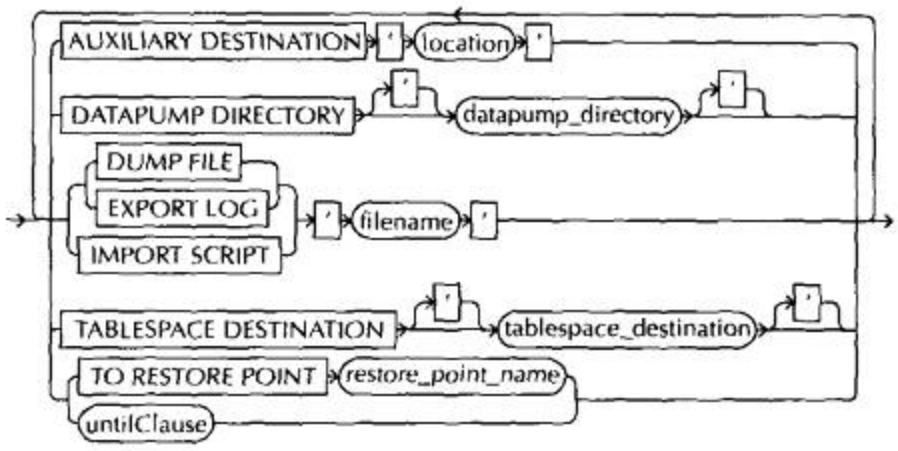


图 A-105　语法图

参考章节

第 14 章详细介绍了可传递的表空间和 RMAN。

示例

```
transport tablespace tbs_2, tbs_3
tablespace destination '/disk1/transportdest'
auxiliary destination '/disk1/auxdest'
UNTIL SCN 251982;
```

A.4.50　unregister database 命令

unregister database 命令删除恢复目录中的数据库和与它相关联的记录，如图 A-106 所示。

语法图

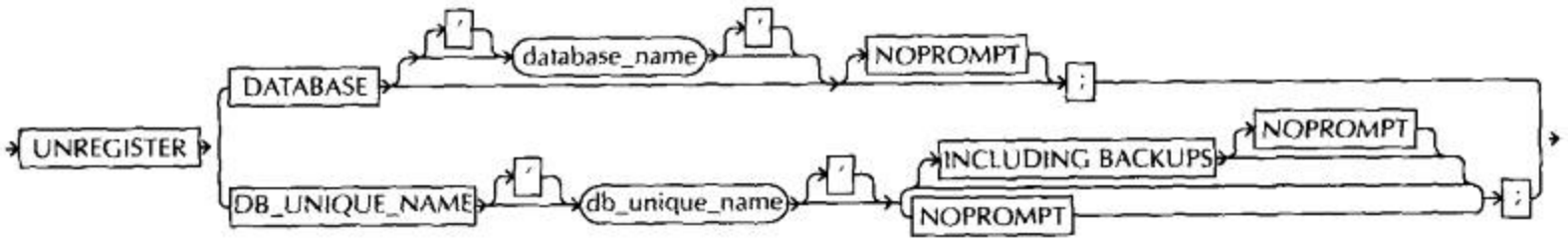


图 A-106　语法图

参考章节

第 3 章详细介绍了 unregister database 命令。

示例

```
unregister database;
unregister database oldprod;
```

A.4.51　upgrade catalog 命令

upgrade catalog 命令用于将恢复目录从低版本升级到更高的 RMAN 版本，如图 A-107

所示。

语法图

图 A-107　语法图

参考章节

第 10 章介绍了 upgrade catalog 命令。

示例

```
upgrade catalog;
```

A.4.52　validate 命令

validate 命令用于检查和验证能否还原给定的备份集，在这个操作期间，RMAN 会扫描所有备份集片，如图 A-108～图 A-113 所示。

语法图

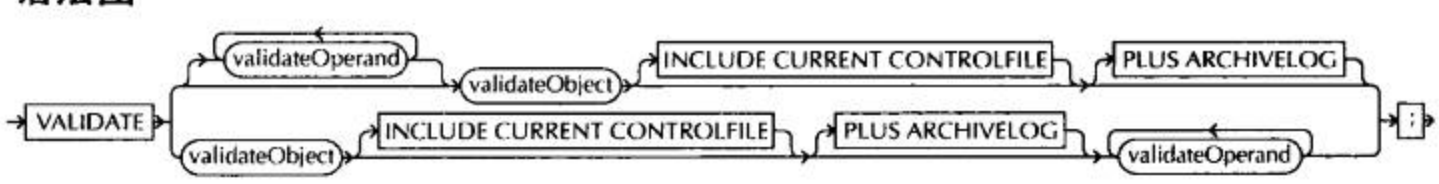


图 A-108　语法图

validateObject 参数语法图

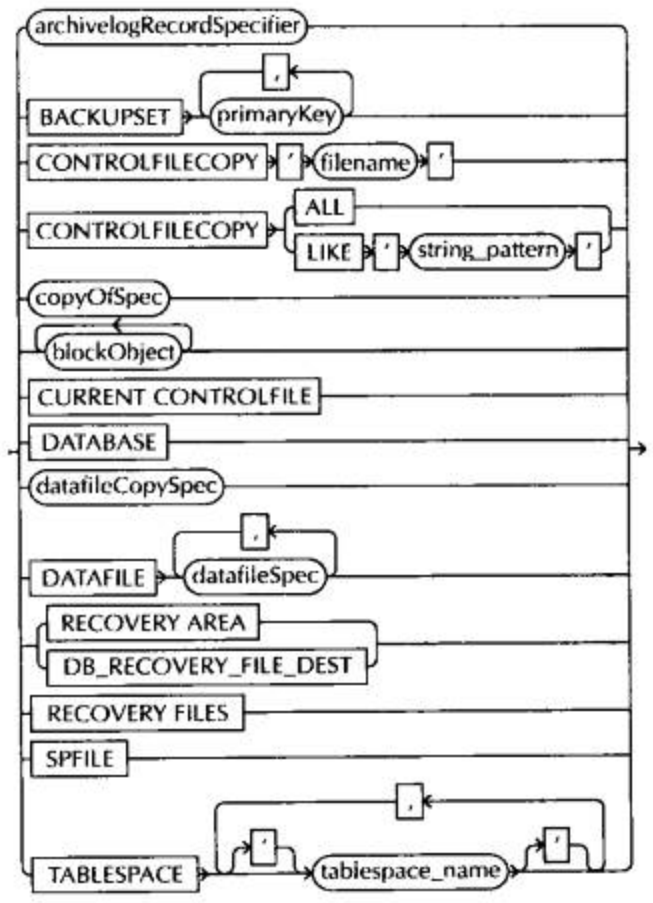


图 A-109　语法图

copyOfSpec 参数语法图

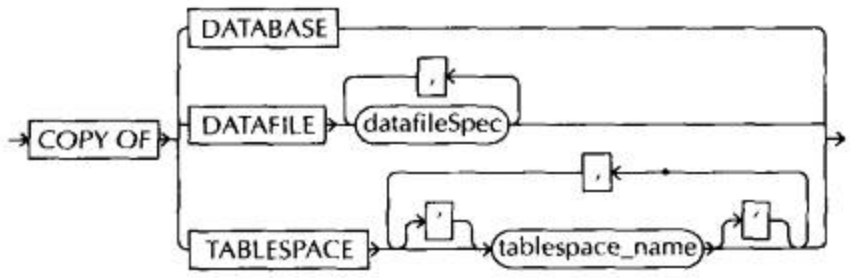


图 A-110　语法图

blockObject 参数语法图

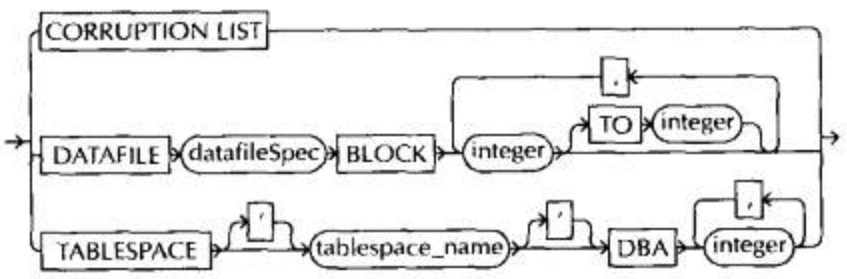


图 A-111　语法图

validateOperand 参数语法图

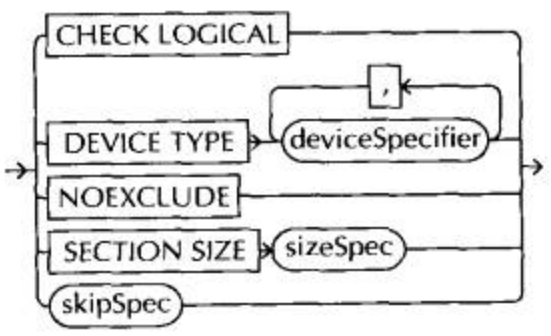


图 A-112　语法图

skipSpec 参数语法图

图 A-113　语法图

参考章节

第 16 章介绍了 validate 命令。

示例

```
validate backup set 1330;
```

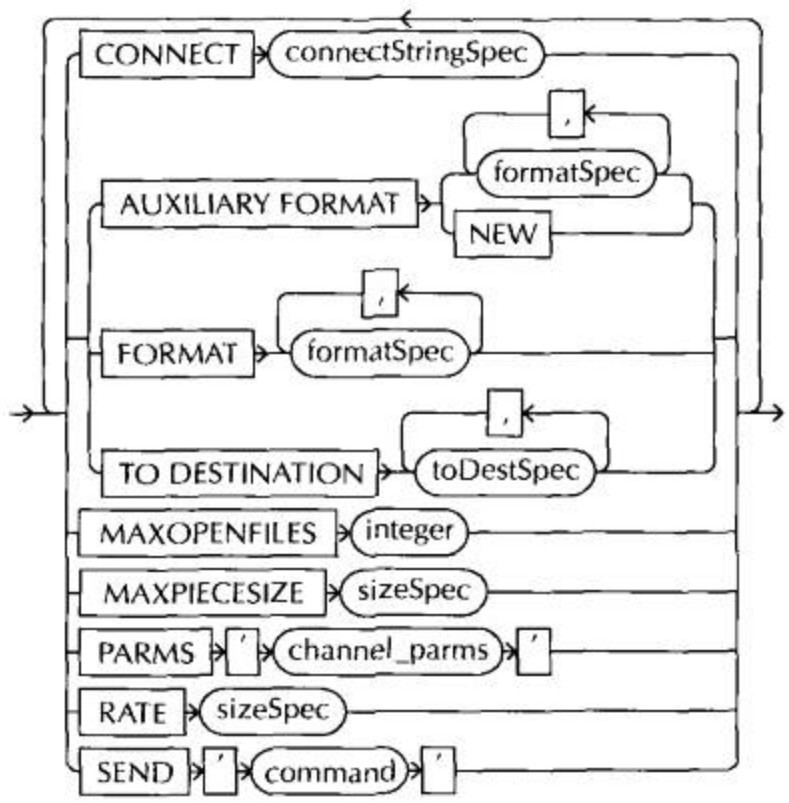
A.5　RMAN子句语法细节

子句出现在许多RMAN命令中，因此在该节中列出它们，以方便用户参考。查看相关命令的部分，了解这些子句用法。

A.5.1　allocOperandList

该操作数指定了与信道相关的控制选项，如图A-114所示。

语法图

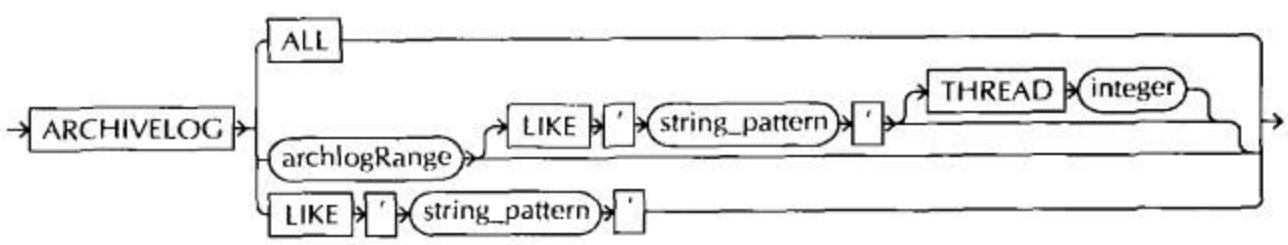


图A-114　语法图

A.5.2　archivelogRecordSpecifier

该子句用于定义恢复周期或在恢复期间应用的归档重做日志，如图A-115～图A-116所示。

语法图



图A-115　语法图

archlogRange 参数语法图

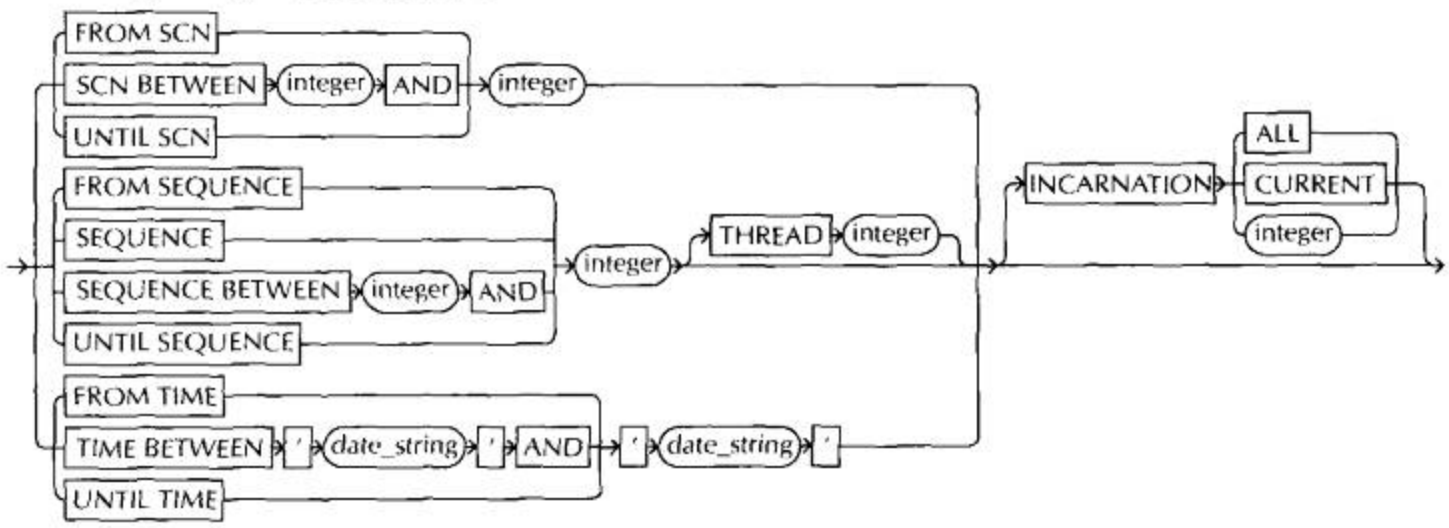


图 A-116　语法图

A.5.3　completedTimeSpec

该命令用于定义备份或副本何时完成。该命令经常与 list、crosscheck 和 delete 命令一起使用，如图 A-117 所示。

语法图

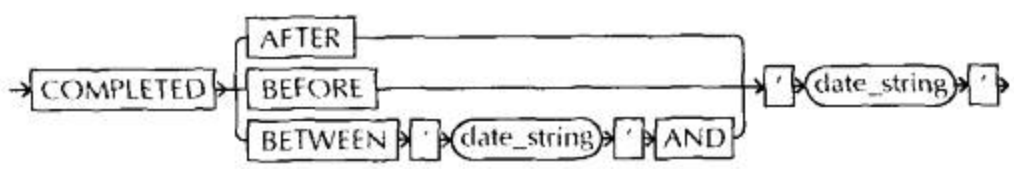


图 A-117　语法图

A.5.4　connectStringSpec

该子句用于定义连接到目标数据库、恢复目录或辅助数据库时的用户名、口令和 Oracle Net 服务名，如图 A-118 所示。

语法图

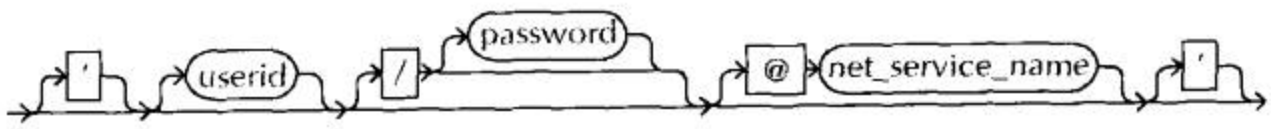


图 A-118　语法图

A.5.5　datafileSpec

该子句用于根据各种操作的文件名或数据文件编号定义数据文件，例如数据文件恢复，如图 A-119 所示。

语法图

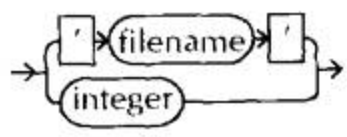


图 A-119　语法图

A.5.6　deviceSpecifier

该子句定义给定的 RMAN 操作使用的存储类型，如图 A-120 所示。

语法图

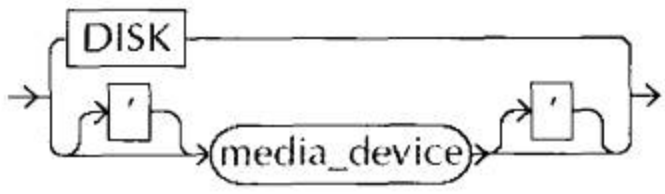


图 A-120　语法图

A.5.7　filNameConversionSpec

该说明符用于定义基于原有数据文件名生成新数据的文件名的模式。查看 backup、convert 和 duplicate 命令的相关小节以了解更多细节，如图 A-121 所示。

语法图

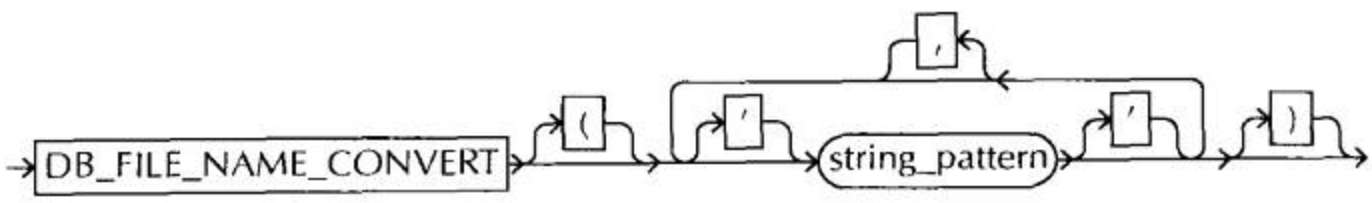


图 A-121　语法图

A 5.8　forDbUniqueNameOption

该子句用于指定是否所有的数据库或者特定的数据库由特定的 RMAN 相关的 Data Guard 操作确定地址，如图 A-122 所示。

语法图

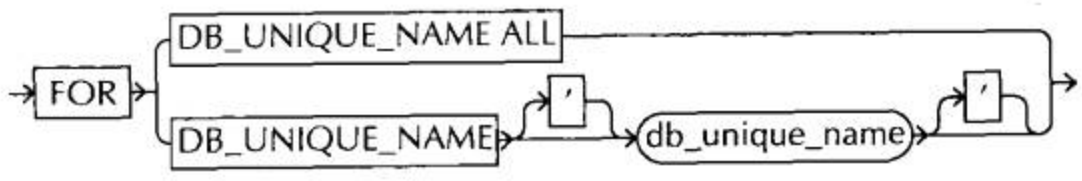


图 A-122　语法图

A 5.9　foreignlogRecordSpecifier

该子句定义了操作中需要使用的特定的外部归档重做日志的集合，如图 A-123 所示。

语法图

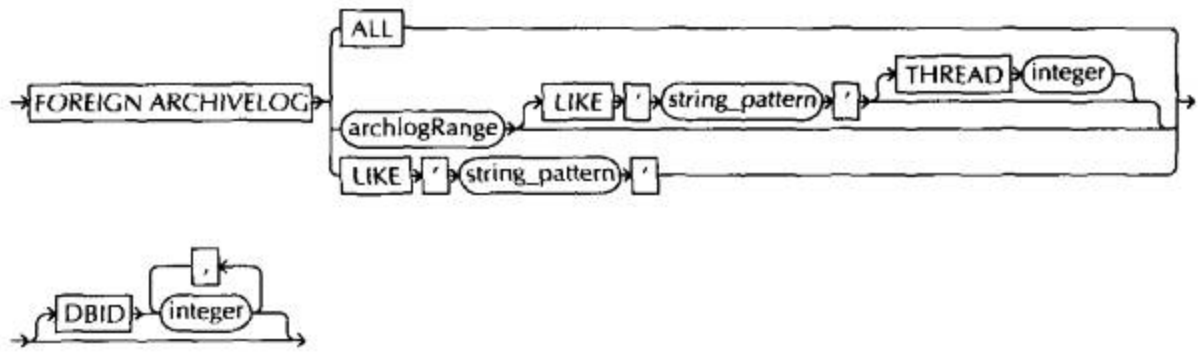


图 A-123　语法图

A.5.10　formatSpec

该说明符用于定义特定的文件名格式，可以包括普通文件名或 ASM 磁盘组，如图 A-124 所示。

语法图

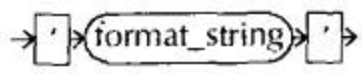


图 A-124　语法图

A.5.11　keepOption

该子句用于定义与给定备份或副本相关的保留策略，如图 A-125 所示。

语法图

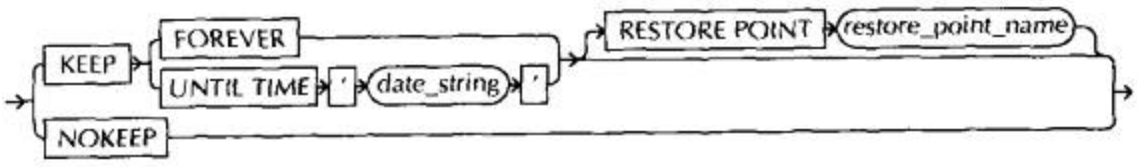


图 A-125　语法图

A.5.12　listObjList

该子句用于指定受父命令影响的特定数据文件和归档重做日志，如图 A-126～图 A-127 所示。

语法图

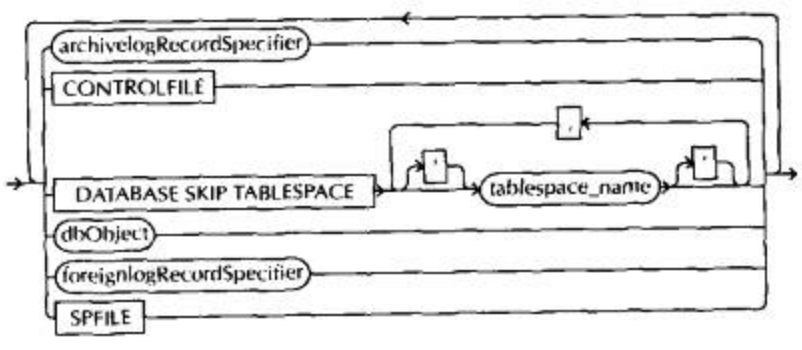


图 A-126　语法图

dbObject 参数语法图

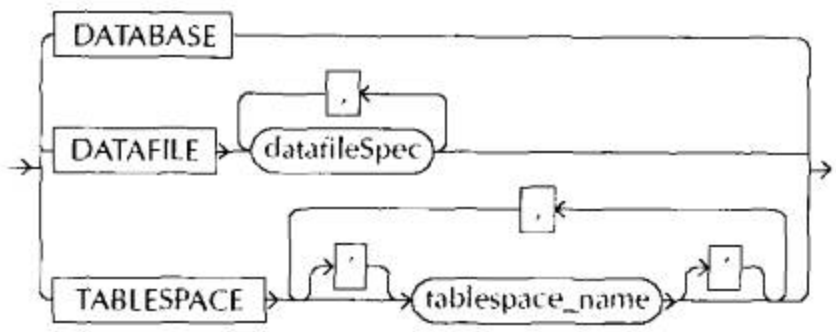


图 A-127　语法图

A.5.13　maintQualifier

该子句用于定定义特定 RMAN 命令中的数据库文件或归档的重做日志，例如 list、crosscheck、delete 和 switch 命令，如图 A-128 所示。

语法图

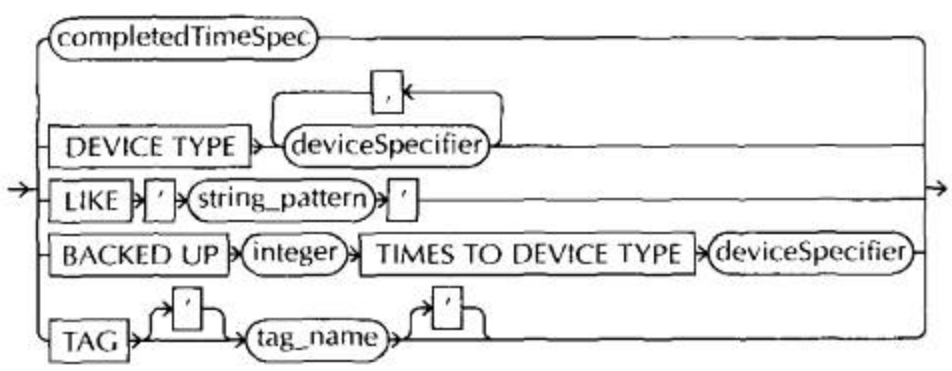


图 A-128　语法图

A.5.14　maintSpec

该子句用于定义特定 RMAN 命令中的数据库文件或归档的重做日志，例如 change、crosscheck 和 delete 命令，如图 A-129 所示。

语法图

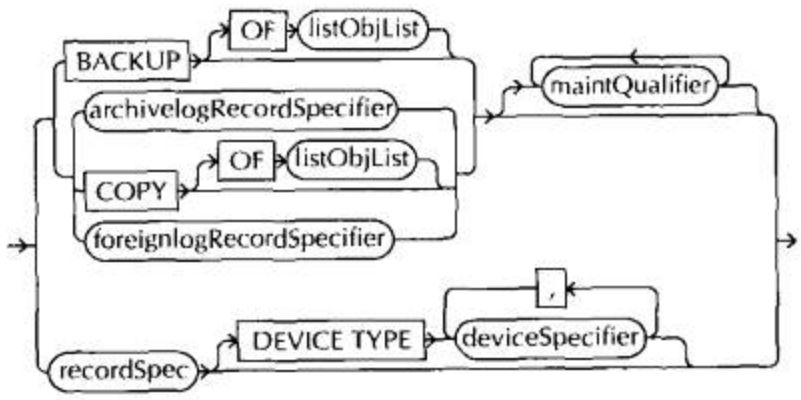


图 A-129　语法图

A.5.15　obsOperandList

该子句用于定义将备份和数据文件副本标记为 OBSOLETE 的标准，如图 A-130 所示。

语法图

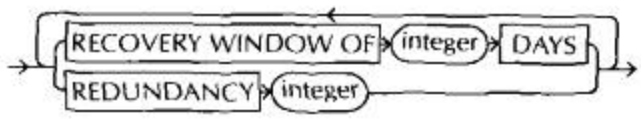


图 A-130　语法图

A.5.16　recordSpec

该子句用于定义 change、crosscheck、delete 和 list 命令将作用的对象，如图 A-131 所示。

语法图

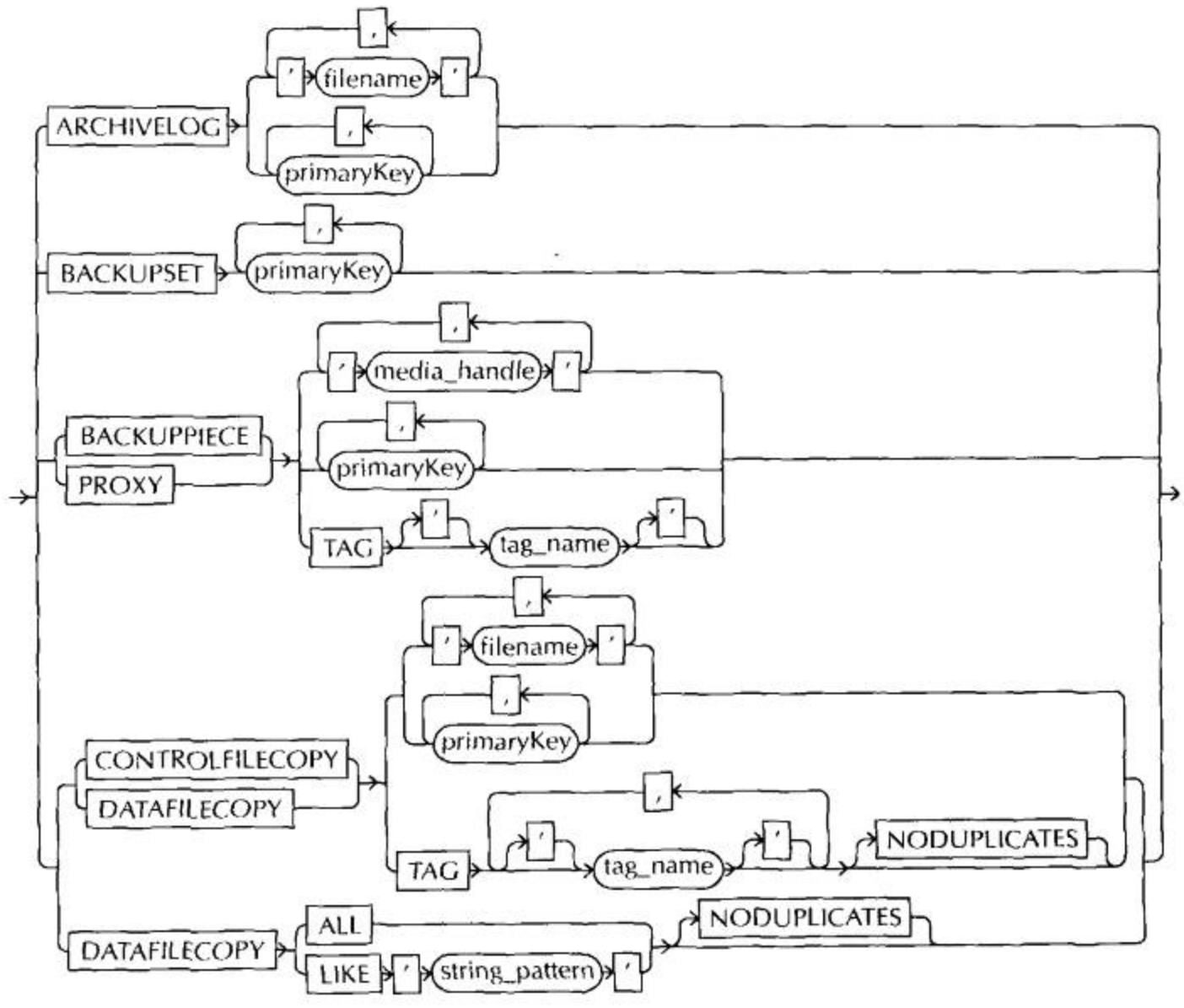


图 A-131　语法图

A.5.17　tempfileSpec

该子句用于按照文件名或文件编号定义应该由父命令操作的临时文件，如图 A-132 所示。

语法图

图 A-132　语法图

A.5.18　toDestSpec

该子句用于为 RMAN 操作的磁盘备份指定目录或自动存储管理磁盘组，如图 A-133 所示。

语法图

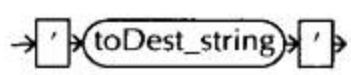


图 A-133　语法图

A.5.19　untilClause

该子句基于时间、SCN、还原点或日志序列号定义由父 RMAN 命令引用的限制，如图 A-134 所示。

语法图

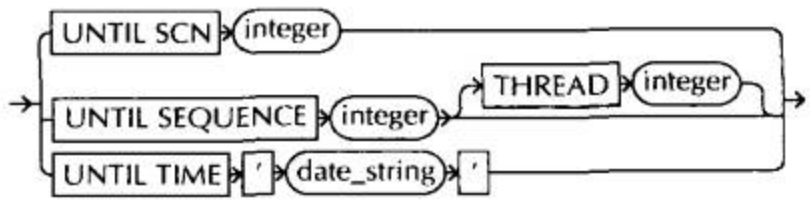
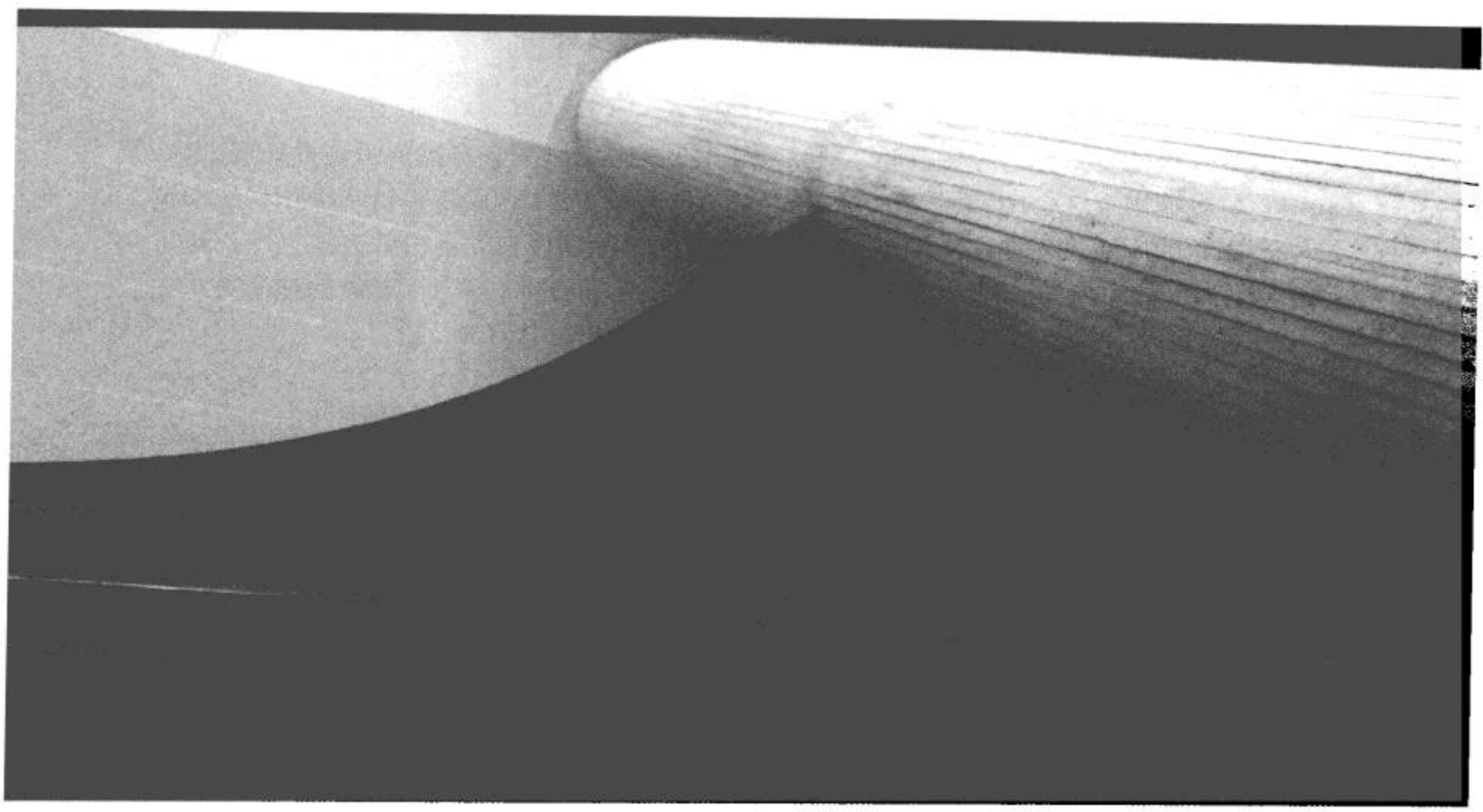


图 A-134　语法图



附录 B

RMAN 脚本示例

我们收到大量请求要求获得与 RMAN 有关的脚本。对于 RMAN 来说好的一方面是针对其撰写脚本是一个非常直接简单的进程。本章将会提供针对 Windows 及 Linux 系统的基础性脚本，以帮助您开始学习脚本编写工作。

这些脚本假定您使用 Oracle 闪回恢复区(FRA，Flash Recovery Area)管理磁盘空间和保留备份。如果未使用 FRA，就需要按照自己的需要定制这些脚本，将这些工作留给用户，通过对脚本的定制，将会增加创造能力和实用技能。

B.1　用于 Windows 的 RMAN 脚本

以下提供的脚本针对 Windows XP 系统编写和测试。首先，将提供示例批处理脚本，利用它可以调用 RMAN 对数据库和归档重做日志进行备份。然后将提供从操作系统中调度这类脚本的方法。需要注意的是本章提供的方法仅是调度自动备份的方法之一。也可以选择使用 Oracle Enterprise Manager(OEM)调度和管理备份。关于 OEM 的使用方法在第 13 章中有详细的介绍，可参考其相关内容实现在 OEM 中调度备份。

B.1.1　创建 Windows 脚本以调度备份

该脚本是很基础的脚本，可能想要对其进行扩展以便应用于增量备份，备份验证等其他操作。应该注意在备份失败时，该脚本将返回一个错误信息。可以使用 Notepad 或其他文本编辑器创建该脚本，可以使用类似 backup.bat 的脚本调用该脚本。

```
rem ************************************************************
rem * Script Name: backup.bat
rem * Script Purpose: This script will call RMAN and execute the command
rem * file specified on the command line.
rem * Usage backup.bat <oracle_sid> <backup|arch>
@echo off
rem
rem RMAN BACKUP SCRIPT
rem For WIN XP
rem
echo %1
set oracle_sid=%1
if "%2" == "backup" rman target=/ cmdfile=c:\oracle\scripts\backup.scr
if not ERRORLEVEL 0 echo "WARNING - FAILURE OCCURRED"
if "%2" == "arch" rman target=/ cmdfile= c:\oracle\scripts\arch.scr
if not ERRORLEVEL 0 echo "WARNING - FAILURE OCCURRED"
```

该脚本调用两个命令文件，分别是 backup.scr 和 arch.scr 脚本，在本例中都位于 c:\oracle\scripts 目录下。以下是 backup.scr 脚本：

```
Backup as compressed backupset database plus archivelog delete input;
```

脚本 arch.scr 如下：

```
Backup as compressed backupset archivelog all delete input;
```

同样，这些脚本可以使用文本编辑器创建并放置在 c:\oracle\scripts 目录下。如果将其放置于其他目录中，则需要修改 backup.bat 脚本，使脚本指向正确的位置。

B.1.2　调度备份

创建脚本后，可以开展备份调度工作。可以使用 Windows 系统的 schtasks 实用程序执行该操作。根据经验，schtasks 是一个很少被使用但具有强大的调度能力的实用程序。下面的示例将使用 backup.bat 文件调度日常数据库备份操作，调度归档重做日志备份操作，以及显示如何删除调度任务。

```
schtasks /create /tn "database_backup" /sc weekly /d SUN /st 14:50:00 /tr
"c:\bc\rman\backup.bat rob10r2 backup>\>c:\bc\rman\backup.output"
rem schtasks /delete /tn "database_backup"
schtasks /create /tn "archivelog_backup" /sc daily /st 14:50:00 /tr
"c:\bc\rman\backup.bat rob10r2 arch>\>c:\bc\rman\backup.output"
```

在使用 schtasks 调度程序时有一些调度选项。使用 schtasks 调度器需要您的登录 ID。

B.2　用于 UNIX 的 RMAN

该类脚本可以在 Red Hat Linux 版本 5 上创建和测试。本节提供备份脚本(backup.ksh)和相关的命令行文件，命令行文件用于执行实际的备份操作。可以在 UNIX 系统中的 Cron or at 调度该脚本。以下是为 Unix 备份的 shell 脚本的示例：

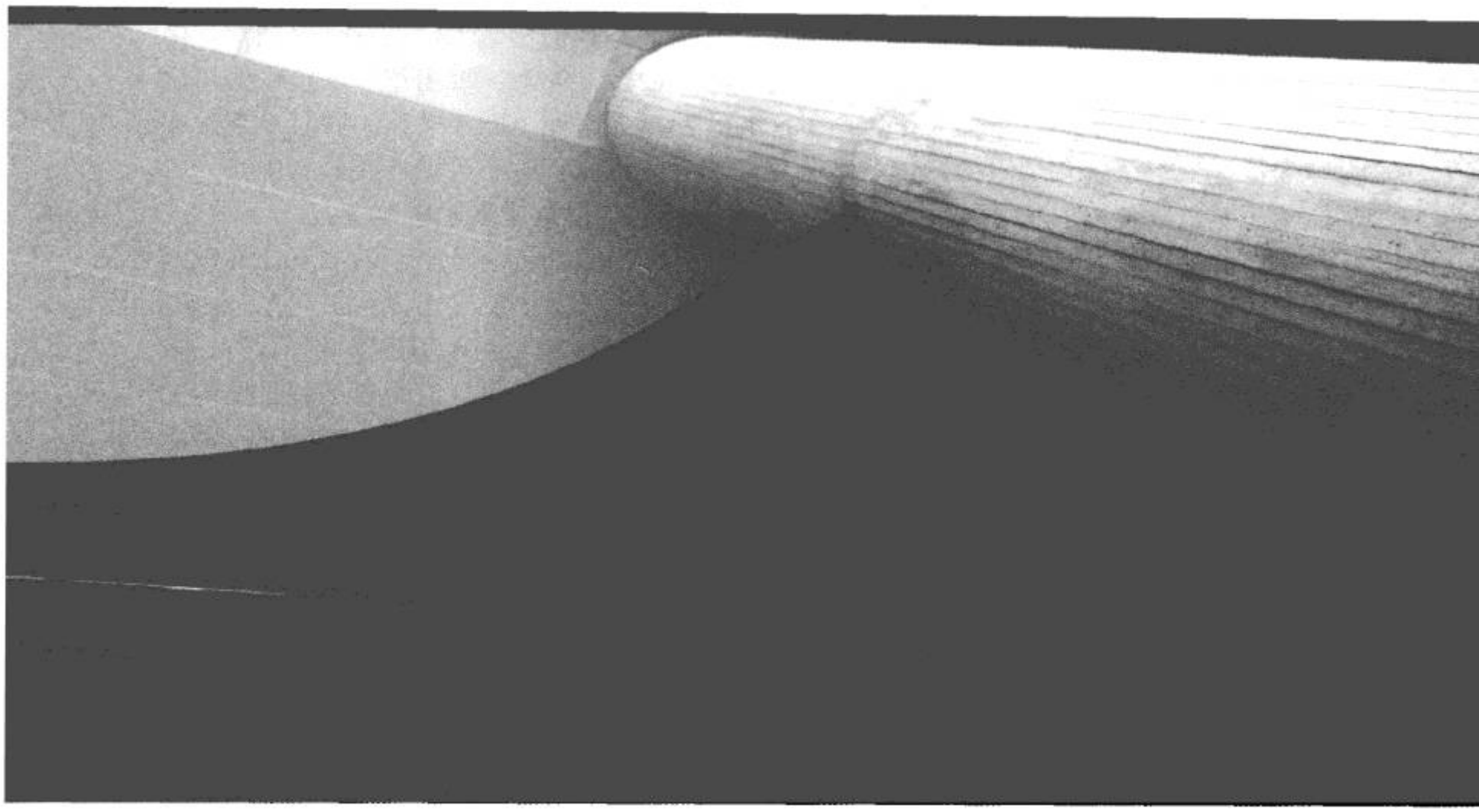
```
#/bin/ksh
# Script name: backup.ksh
# Usage: backup.ksh <ORACLE_SID> <backup|arch>
# Note: We assume the oracle environment is already setup except for
# ORACLE_HOME. If not, you will need to setup your environment correctly.
set ORACLE_SID=$1
if [ "$2"="backup" ]; then
rman target=/ cmdfile=/home/oracle/scripts/backup.scr
fi
if [ "$2"="arch" ]; then
rman target=/ cmdfile=/home/oracle/scripts/arch.scr
fi
```

代码中的 backup.scr 脚本与上节所示的相同：

```
Backup as compressed backupset database plus archivelog delete input;
```

代码中的 arch.scr 脚本与上节所示的相同：

```
Backup as compressed backupset archivelog all delete input;
```



附录 C

创建 RMAN 测试环境

随着产品的企业环境变得越来越复杂，DBA 发现这种复杂性也带入到了测试环境中。例如，在清华大学出版社引进并出版的《Oracle 9*i* RMAN 备份与恢复技术》中，测试环境表面上看起来非常复杂：一个用于较少调整和屏幕截图的 Windows 笔记本电脑、一个用于较多可靠性测试的 Windows 服务器以及用于多个 OS 交互和 Unix 命令的 Sun Blade 150。在这 3 台机器之间，我们能够执行所有的技术评审(当然，这需要结合多年的实际工作经验)。

对于 10*g* RMAN 版本的书籍来说，测试环境包括具有共享的 FireWire 磁盘驱动器(当然是运行 RAC)的两台 Linux 机器、可信任的 Windows 笔记本电脑、原有的 Sun Blade 以及一台单独的 Linux 机器。我们仍然需要和同事一起查找其他的 RAC 集群、磁带存储光盘机以及 Oracle 企业管理器储存库。

尽管如此，与上一版书籍比较，本书在测试环境方面仍然做了一些有用的调整。由于本书的作者们在撰写本书的过程中还有其他工作要做，经常需要出差。从策略的角度来看，其需求也发生了显著的变化。由于 B&R 测试在时间上断断续续，并且不断修改反复的特性，对于远程节点的测试非常困难。在乘飞机旅行时不可能连接数据中心。另外，由于是首次使用 Beta 代码，因此需要花费一些时间以适应新的解决方案。

本书的测试环境如何呢？使用新型的 Apple MacBook Pro,运行 VMWare Fusion 以及一个外接硬盘驱动器，仅此而已。(当然，Matthew 仍然需要数据中心的那些 RAC 集群)他安装了 RHEL5，复制了部分 VMWare，使其便携式电脑能够在多服务器环境运行，多么现代的时代。至此，Matthew 拥有了一个完整的移动试验环境，有足量的硬盘空间和内置方式用以删除无用的垃圾并重新开始使用虚拟快照。唯一的限制是其最新的 MAC 仅有 3GB 内存，因此在运行多服务器环境时需要仔细选择环境。如果新环境能够升级到 8GB 内存，则可以消除上述缺陷。

当然，正如后续附录中所讨论的情况，这样的环境仍然与真实的产品环境有一些差距。因此，在此测试环境中，Matthew 仍然不能完成所有的性能测试，或者不能保证其脚本在实际环境中没有错误。使用运行(RHEL)良好的操作系统和数据库版本，有利于帮助 Matthew 确信他执行的操作及这些操作实现的方式。

我们在清华大学出版社引进并出版的《Oracle 9*i* RMAN 备份与恢复技术》一书中声明过，需要一定的技巧来描述测试环境或提供关于测试环境的建议。每个商店都对应用程序设计、质量保证、版本控制等方面有不同的测试需求和测试级别。随着 RMAN 在 DBA 的各种活动中扮演着更为重要的角色，我们更加难以将 RMAN 测试环境与其他测试环境分离开来。但是，通过阅读本书，就可以了解测试备份策略。每个人都有备份策略，但却很少有人有恢复策略。测试备份只是工作的一小部分，如果没有测试恢复策略，则无论将多少文件写入到价格昂贵的设备中，都不会有任何备份。

用于备份与恢复的测试环境不同于其他测试环境。首先，必须能够删除数据文件甚至整个数据库，而不用事先通知其他用户。也就是说，需要个人使用的数据库。如果在共享的数据库上开始测试 RMAN 功能，其他用户很快会投诉这种行为，或者您会被 SA 锁定在机器之外。备份与恢复测试环境很容易变得不稳定，因而不能被共享。可以设想另一种情况：您正忙于测试自己的备份操作，而另一位用户为了测试自己的还原与恢复操作而删除了数据文件，这样您的备份操作就会异常中断。

另一方面，还需要在与产品数据库非常相匹配的环境中测试您的策略。由于可能需要在具有与产品数据库相同负载的大型产品级服务器上调整备份操作，因此不能总是独立地运行各种操作。

因此，我们建议从两个层次研究 RMAN 备份与恢复测试。首先，在一个独立的小型测试服务器上熟悉 RMAN 的功能和行为；然后，在一个大型的产品级数据库服务器上运用在第一步中积累的经验来进行测试。这样，利用一台测试机器就能完成备份/恢复测试，而不必非得在大型工作站上完成。

接下来，本附录将介绍这些方法。

C.1　测试机器

测试 RMAN 功能的第一级机器不必是超型计算机。实际上，第一级测试仅仅是预演，此时只需要通读命令行，确保各项部署正确并与相关人员就各步骤逐项进行了讨论。

C.1.1　匹配产品环境

如果可能的话，进行 RMAN 测试的操作系统应当和运行产品数据库的操作系统相同。不过这是不现实的，用户环境中已经不仅仅只存在单个操作系统了。然而，如果只想在 Solaris 服务器上执行备份操作，那么花点儿钱购买 Sun 工作站是合情合理的，这样用户才可能尽快地开始匹配产品环境。

C.1.2　节约成本

是否在与产品环境相同的操作系统上进行第一级测试并不重要。RMAN 在所有平台上的作用是相同的，本书中的所有练习可以在所有平台上运行。因此，如果您负责购买 RMAN 测试机器，我们的建议是：节约成本。可以购买一台运行 Windows 或 Linux 的普通计算机，可以购买二手机而不必购买配置很高的机器。我就非常喜欢自己的二手 Linux 集群，清华大学出版社引进并出版的《Oracle Database 10g 高可用性实现方案——运用 RAC、Flashback 和 DataGuard 技术》对此进行了介绍。这是带有两个节点的 RAC 测试机器，其价格在 1500 美元以下，由 Dell 经过整修后出售。如果您正在寻找廉价的测试环境，下面是我们提供的建议。

- 处理器速度

出于 RMAN 概念证明的目的，不必考虑处理器速度。RMAN 不是特别依赖于 CPU，但如果转移到并行化程度很高的产品环境，CPU 速度对于测试就非常重要。即使此时监控了性能，可是与产品环境比较时，这些测试数据仍然是无意义的。相反，我们应当将资金用于其他资源，其中最主要的资源是磁盘空间和内存。

- 内存

需要配置足够的内存以同时运行三个 Oracle 实例和介质管理软件。如果要使用 OEM 或其他管理软件包进行测试，仍要考虑这些因素。我们建议至少配置 2GB 内存，否则，就会启动内存交换，运行速度则会大大降低。

- 磁盘空间

磁盘比较便宜，不必考虑使用多个 SCSI 磁盘，而只需要考虑磁盘的空间。同样，速度在这一级测试中并不重要。购买一个 200GB 的硬盘驱动器，这个容量应当足够使用。这一级别进行的是概念测试，因此可以控制数据库的大小，以保证控制所有操作。不过需要记住的是，如果存在多个数据库并且都要被备份到磁盘上，就需要为 RMAN 备份片分配空间。因此，应尽可能使用更大的磁盘空间。

如果决定执行 RAC 测试程序，则除了在机器内部加载的硬件外，还需要外部的 SCSI 或 FireWire 驱动器。可参考清华大学出版社引进并出版的《Oracle Database 10g 高可用

性实现方案——运用 RAC、Flashback 和 Data Guard 技术》来进一步了解 RAC 主集群。

C.2 Oracle 配置

准备好并运行测试机器后，就需要考虑 Oracle 的安装和配置。这一步骤取决于实际的测试要求，例如是否要备份多个 Oracle 版本，是否要使用 OEM(Oracle Enterprise Manager)？

C.2.1 多个主目录

如果要检验多个 Oracle 版本，就务必要按照由低到高的版本顺序安装这些版本。例如，在 10.1.0 版本前安装 9.2.0 版本，在 10.2.0 版本前安装 10.1.0 版本。在执行其他操作前，需要为 Oracle 添加最新的补丁，这样通常能够修复新发现的 RMAN 程序故障。

C.2.2 创建数据库

显然，至少需要为每个已经安装的 ORACLE_HOME 创建一个数据库。这些数据库可能是在 Oracle 安装期间创建的默认数据库，但是最好使用其配置与产品数据库的配置接近的数据库。显然，从容量的角度出发，配置相同大小的数据库不大可能，但是可以在保持数据文件和表空间数目相同的情况下缩小数据文件的大小。

此外，应尽可能地减少测试机器的内存使用。实际上，不会执行太多的操作，因此不需要使用太多的缓冲区缓存。系统全局区(SGA)越小，则测试机器的性能就越好。

为了进行测试，除了使用目标数据库之外，我们还需要一个恢复目录数据库。通常，我们建议让恢复目录数据库使用最新的版本，所以应当将这个数据库放入 11.2 版本的主目录中。在必要时，恢复目录数据库也可以被用作目标数据库，不过尽量将恢复目录数据库放在要删除和重新构建的数据库之外，这样会使操作更加简单。如果可能的话，可以将恢复目录数据库放在不同的服务器上。并将其放入 Windows 工作站或原有的 Linux 机器。总之，要确保将其与最终会删除的路径分离开来。

C.2.3 使用 Oracle ASM

如果计划测试 Oracle 的卷管理器 ASM(自动存储管理)，则在第一次配置 RMAN 测试机器时必须为此做好准备。在产品环境中，只需要向 ASM 磁盘组添加完整的、未经过处理的原始磁盘。在测试环境中，如果需要测试多个 ASM 磁盘组，则可以使用一个磁盘上的逻辑分区。但是这意味着必须提前考虑到这一点，要尽量在进行 OS 设置之前在磁盘上创建一些未使用的、未经过处理的分区。

C.2.4 Oracle Enterprise Manager

如果计划使用 Oracle Enterprise Manager，就需要确保有足够的内存。在第 13 章中介绍过，可以选择两种 OEM 方式：Database Control 和 Grid Control。从测试的角度来看，使用 Database Control 更有意义，它可以节省资源并避免管理问题。然而，需要根据是否匹配产品环境来选择：如果在产品中部署 Grid Control 管理，则在测试环境中使用 Grid

Control 管理 RMAN 备份。

这也就是说，应尽量避免将产品 Grid Control 用于 RMAN 测试环境。数据库将会停机、启动、停机、丢失、废弃、崩溃和丢失。这意味着许多警报会发送给 Oracle 管理服务(Oracle Management Service，简写为 OMS)，并且由此发送电子邮件给用户。我们应避免产生大量这些无意义的电子邮件！如果没有为企业部署测试 Grid Control 环境，则为 RMAN 备份创建一个测试 Grid Control 环境，然后出于测试目的将其提供给其他备份。

对于 Database Control，每个实例要占用 150MB 到 200MB 的内存。同样，它对 CPU 的使用也很频繁，并且用尽数据库空间。对于空闲的数据库，Database Control 每天生成大约 200MB 的归档日志。

对于 Grid Control，总共需要 2GB 的系统用于存储库数据库和 OMS，并且将其作为独立的系统包括进来。在 RMAN 测试系统上，代理将只使用大约 60MB 的内存和足够的 CPU 来运行它的 Perl 脚本。

无论是 Database Control 或 Grid Control，OEM 都与稳定的、可预测的互联子层高度相关，这意味着不可以经常更改主机名或 IP 地址。如果必须更改，则创建自己的子网并在主机文件中手动分配虚拟 IP 地址。确保所有方面将正确运行的最简单方法如下：在主机名上使用 nslookup 命令，然后在 IP 地址上进行反向查找。

C.2.5　介质管理问题

如果可以的话，应当在产品数据库中安装要使用的介质管理客户端版本，随后还应当安装 Oracle 插件，并且像产品数据库那样在磁带上生成备份。这样就对自己在企业中实现的策略有了最佳的预见力。

如果不能访问用于企业的介质管理产品，则只有很少的选择，建议安装第 5 章中介绍的 Oracle Secure Backup。

如果只需要测试磁带信道分配，或者是测试将闪回恢复区分段传输到磁带的进程，则需要使用 Oracle SBT API，这种 API 允许将“磁带”备份写入到磁盘设备，在第 4 章的 RMAN 实习“使用 Oracle 默认的 SBT 界面测试磁带信道”中对此进行了介绍。

C.3　RMAN 配置

现在，系统中已安装了 Oracle 并创建好了数据库，此时还需要注意下面这些与测试进程有关的问题：

- 保留一个不受影响的冷备份。执行 RMAN 测试前，关闭数据库并生成一个 OS 副本冷备份，然后将这个备份放入一个不受影响的文件夹中。如果在 RMAN 测试进程中完全陷入困境，这将是您的最后一条防线。
- 多次切换重做日志。RMAN 测试会引起的一个最大错误在于备份与还原之间的时间帧非常短。由于备份操作的完成时间和还原操作的“until time”之间没有间隔，所以很容易引起混淆。因此，在备份操作之后，最好确保切换 3 次或 4 次日志文件，这样就能够在备份操作和还原操作之间设置可区分的间隔。

- 设置 NLS_DATE_FORMAT 环境变量。对于 RMAN 来说，这一般是很好的建议，但在测试情况中特别有用。在进行测试时，备份和还原之间的时间帧将非常短，并且需要知道精确到秒的备份时间帧。因此，在启动 RMAN 之前，确保运行如下命令：

```
export NLS_DATE_FORMAT='mon-dd-yyyy hh24:mi:ss'
```

然后，在启动 RMAN 并执行 list backup 命令时，将总是显示精确到分钟和秒的详细时间。

- 不要使用目录数据库。我们可能会将该目录数据库作为目标数据库来进行一些测试，这没有问题，要不然怎么叫测试环境呢。但是，如果损坏了恢复目录，就会严重地破坏 RMAN 测试。我们建议尽量在专用位置保存恢复目录数据库，然后在开始新的测试会话前使用用户级输出导出这个恢复模式。
- 随时进行目录维护。虽然这是一个测试环境，但是我们会不断地创建备份，而测试机器的空间是有限的。可以利用这个机会测试使用保留策略删除旧的备份。
- 尽快删除副本。如果使用了 duplicate 命令，就会运行许多不同的实例，这将耗尽宝贵的内存和磁盘空间资源。对于测试环境中的这些副本，一定要尽快删除。
- 保留复制的文件系统。不必在每次测试 duplicate 或 duplicate for standby 命令时都执行一遍为数据库副本构建文件系统和 init.ora 文件的步骤。保持文件系统和支持文件不变，并且使用相同的 DB_NAME 和 SID。在 Windows 平台上，在 Services 控制面板上保持 Oracleservice<SID>不变。
- 不要局限于测试环境。有时，特别是在没有良好的维护习惯的情况下，我们需要删除所有文件并从头再来。经过多次测试后，数据库肯定会进入混乱状态：表空间被删除、重新创建、删除文件，且使不常使用的文件被置入错误的目录，归档日志被到处存放。不要着急，测试就是这样。对测试环境不必留情，把所有的东西都删掉，我们可以用在测试前生成的冷备份从头开始。

在完成一些测试后，一定会有所收获。用测试机器实际地理解了一些概念后，可以利用创建的脚本和获得的知识在企业级系统上做一些练习，以确保在企业的数据库上一切都将正常运行。

[General Information]
书名=Oracle Database 11g RMAN备份与恢复
作者=（美）弗里曼,（美）哈特著
页数=612
出版社=北京市：清华大学出版社
出版日期=2011.04
SS号=12773610
DX号=000008084076
URL=http://book.szdnet.org.cn/bookDetail.jsp?dxNumber=000008084076&d=F881C393E681C2694F95B707431 8F586