VERITAS NetBackup™ 4.5 for Oracle ServerFree Agent

System Administrator's Guide

Solaris and HP

VERITAS

Contents

Preface

This guide describes how to install, configure and use VERITAS NetBackup for Oracle ServerFree Agent on a UNIX platform. In this guide, VERITAS NetBackup for Oracle on UNIX ServerFree Agent is referred to as either NetBackup for Oracle or as NetBackup for Oracle ServerFree Agent.

For specific information about the NetBackup server software, refer to *NetBackup System Administrator's Guide - UNIX* server. For specific information about the NetBackup ServerFree Agent, refer to *NetBackup ServerFree Agent System Administrator's Guide - UNIX* server.

This document is the same as `NetBackup_AdminGuide_ServerFree_Oracle.pdf` distributed with the NetBackup for Oracle ServerFree Agent software.

Audience

This guide is intended for the:

◆ Oracle database system administrator responsible for configuring and using the Oracle Recovery Manager to back up and restore Oracle databases.

◆ NetBackup system administrator responsible for configuring NetBackup.

A system administrator is defined as a person with system administrator privileges and responsibilities.

This guide assumes:

◆ A basic understanding of system administration.

◆ A working understanding of the NetBackup client and server software.

◆ A familiarity with the information covered in the following NetBackup manuals:

  - *NetBackup User's Guide - UNIX*

  - *NetBackup System Administrator's Guide - UNIX*

  - *NetBackup ServerFree Agent System Administrator's Guide*

  - *NetBackup Troubleshooting Guide - UNIX*

◆ A thorough understanding of Oracle Recovery Manager.

Accessibility

NetBackup contains features that make the user interface easier to use by people who are visually impaired and by people who have limited dexterity. Accessibility features include:

◆ Support for assistive technologies such as screen readers and voice input (Windows servers only)

◆ Support for keyboard (mouseless) navigation using accelerator keys and mnemonic keys

For more information, see the NetBackup system administrator's guide.

Organization

This guide is organized as follows:

◆ The first chapter in this manual describes the technical features and concepts of NetBackup for Oracle.

◆ The Installation chapter explains how to install NetBackup for Oracle ServerFree Agent on your system.

◆ Configuration describes how to configure NetBackup and the Oracle Recovery Manager for use with NetBackup for Oracle ServerFree Agent. This information supplements the NetBackup administration manuals.

◆ The Using NetBackup for Oracle ServerFree Agent chapter explains how to use NetBackup to perform backups and restores of Oracle databases. This information supplements the NetBackup manuals.

◆ Troubleshooting explains how to use NetBackup logs to troubleshoot NetBackup for Oracle ServerFree Agent operations. This information supplements the NetBackup manuals.

◆ Appendix A provides examples of the NetBackup for Oracle scripts provided with this product.

Related Documents

The following documents provide related information. For a more detailed listing of NetBackup documents, refer to *NetBackup Release Notes*.

◆ *NetBackup System Administrator's Guide - UNIX*

Explains how to configure and manage NetBackup on a UNIX system.

◆ *NetBackup ServerFree Agent System Administrator's Guide - UNIX*
NetBackup for EMC Fastrax System Administrator's Guide - UNIX

Explains how to configure and manage the NetBackup ServerFree agent.

◆ *NetBackup Media Manager System Administrator's Guide - UNIX*

Explains how to configure and manage the storage devices and media on UNIX NetBackup servers. Media Manager is part of NetBackup.

◆ *NetBackup Troubleshooting Guide - UNIX*

Provides troubleshooting information for UNIX-based NetBackup products. You can also refer to www.veritas.com knowledge base for tech notes.

For this product, also refer to the following Oracle manuals:

◆ *Oracle Enterprise Manager Administrator's Guide*

◆ *Oracle Backup and Recovery Guide*

◆ *Oracle Server Administrator's Guide*

◆ *Oracle Recovery Manager User's Guide and Reference*

Conventions

The following explains typographical and other conventions used in this guide.

Type Style

Typographic Conventions

Typeface	Usage
Bold fixed width	Input. For example, type **cd** to change directories.
Fixed width	Paths, commands, filenames, or output. For example: The default installation directory is /opt/VRTSxx.
Italics	Book titles, new terms, or used for emphasis. For example: *Do not* ignore cautions.
Sans serif (italics)	Placeholder text or variables. For example: Replace *filename* with the name of your file.
Serif (no italics)	Graphical user interface (GUI) objects, such as fields, menu choices, etc. For example: Enter your password in the **Password** field.

Notes and Cautions

Note This is a Note. Notes are used to call attention to information that makes using the product easier or helps in avoiding problems.

Caution This is a Caution. Cautions are used to warn about situations that could cause data loss.

Key Combinations

Some keyboard command sequences use two or more keys at the same time. For example, holding down the **Ctrl** key while pressing another key. Keyboard command sequences are indicated by connecting the keys with a plus sign. For example:

Press Ctrl+t

Command Usage

The following conventions are frequently used in the synopsis of command usage.

brackets [ ]

The enclosed command line component is optional.

Vertical bar or pipe (|)

Separates optional arguments from which the user can choose. For example, when a command has the following format:

command *arg1*|*arg2*

the user can use either the *arg1* or *arg2* variable.

Terms

The terms listed in the table below are used in the VERITAS NetBackup documentation to increase readability while maintaining technical accuracy.

Term	Definition
Microsoft Windows, Windows	Terms used as nouns to describe a line of operating systems developed by Microsoft, Inc.
	A term used as an adjective to describe a specific product or noun. Some examples are: Windows 95, Windows 98, Windows NT, Windows 2000, Windows servers, Windows clients, Windows platforms, Windows hosts, and Windows GUI.
	Where a specific Windows product is identified, then only that particular product is valid with regards to the instance in which it is being used.
	For more information on the Windows operating systems that NetBackup supports, refer to the VERITAS support web site at http://www.support.veritas.com.

Term	Definition (continued)
Windows servers	A term that defines the Windows server platforms that NetBackup supports; those platforms are: Windows NT and Windows 2000.
Windows clients	A term that defines the Windows client platforms that NetBackup supports; those platforms are: Windows 95, 98, ME, NT, 2000, XP (for 32- and 64-bit versions), and LE.

Getting Help

For updated information about this product, including system requirements, supported platforms, supported peripherals, and a list of current patches available from Technical Support, visit our web site:

    http://www.support.veritas.com/

VERITAS Customer Support has an extensive technical support structure that enables you to contact technical support teams that are trained to answer questions to specific products. You can contact Customer Support by sending an e-mail to support@veritas.com, or by finding a product-specific phone number from the VERITAS support web site. The following steps describe how to locate the proper phone number.

1. Open http://www.support.veritas.com/ in your web browser.

2. Click **Contact Support**. The *Contacting Support Product List* page appears.

3. Select a product line and then a product from the lists that appear. The page will refresh with a list of technical support phone numbers that are specific to the product you just selected.

Introduction 1

NetBackup for Oracle on UNIX integrates the database backup and recovery capabilities of Oracle Recovery Manager with the backup and recovery management capabilities of NetBackup and its Media Manager.

This chapter introduces NetBackup for Oracle ServerFree Agent and how it relates to Oracle Recovery Manager and NetBackup. Read this chapter for a description of:

◆ Features of NetBackup for Oracle on UNIX

◆ Terminology for NetBackup for Oracle on UNIX

◆ Technical Overview of NetBackup for Oracle on UNIX

In this guide, VERITAS NetBackup for Oracle on UNIX ServerFree Agent is referred to as NetBackup for Oracle on UNIX or as NetBackup for Oracle ServerFree Agent.

Features of NetBackup for Oracle on UNIX

This section describes the NetBackup for Oracle on UNIX main features.

Feature	Description
Media and device management	All devices supported by Media Manager are available to NetBackup for Oracle on UNIX.
Scheduling facilities	NetBackup scheduling facilities on the master server can be used to schedule automatic and unattended Oracle backups.
	This also lets you choose the times when these operations can occur. For example, to prevent interference with normal daytime operations, you can schedule your database backups to occur only at night.
Multiplexed backups and restores	NetBackup for Oracle on UNIX lets you take advantage of NetBackup's multiplexing capabilities. Multiplexing directs multiple data streams to one backup device, thereby reducing the time necessary to complete the operation.
Transparent execution of both Oracle and regular file system backup and restore operations	All backups and restores are executed simultaneously and transparently without any action from the NetBackup administrator.
	A database administrator can execute database backup and restore operations through NetBackup or use Oracle Recovery Manager as if NetBackup were not present.
	An administrator or any other authorized user can use NetBackup to execute database backups and restores.
Sharing the same devices and tapes used during other file backups	It is possible to share the same devices and media used for other backups or to give Oracle exclusive use of certain devices and media.
Centralized and networked backup operations	From the NetBackup master server, you can schedule database backups or start them manually for any client. The Oracle databases can also reside on hosts that are different from the devices on which NetBackup stores the backups.

Feature	Description
Graphical user interfaces	NetBackup provides the following graphical user interfaces for client users and administrators: ◆ Backup, Archive, and Restore user interface ◆ Client user motif interface, xbp ◆ NetBackup Administration Console for Java ◆ NetBackup Administration Console for Windows A database administrator or NetBackup administrator can start backup or restore operations for Oracle from the NetBackup graphical user interface on the master server.
Parallel backup and restore operations	NetBackup for Oracle on UNIX supports the parallel backup and restore capabilities of the Oracle Recovery Manager. This permits the user to run more than one tape device at a time for a single Oracle backup or restore, thereby reducing the time necessary to complete the operation.

Terminology for NetBackup for Oracle on UNIX

This section explains important terms that may be new to an Oracle database administrator or a NetBackup administrator.

NetBackup Terms

This section describes NetBackup terms as they apply to NetBackup for Oracle on UNIX.

NetBackup	NetBackup backs up and restores files, directories, raw partitions, and databases on client systems that have Oracle databases.
NetBackup policy	NetBackup policies define criteria for backups and restores. These criteria include storage unit and media to use, backup schedules, clients, files to back up, or backup templates or scripts to execute.
NetBackup schedule	NetBackup schedules control NetBackup operations such as: when backups can occur, the type of backup (full, incremental) to perform, and how long NetBackup retains the image (retention level).
Administrator directed backups	NetBackup administrators are able to perform remote backups of all files, directories, databases, and raw partitions contained on client systems within a client/server network via NetBackup interfaces.
User-directed backups and restores	NetBackup users are able to perform backups of all files, directories, databases, and raw partitions contained on client systems within a client system.
Graphical interfaces	Graphical user interfaces are available for both users and administrators.
Media Manager	The Media Manager provides complete management and tracking of all devices and media used during backups and restores.
Templates	The NetBackup for Oracle wizards are used to create backup and recovery templates. These wizards are initiated from the NetBackup Backup, Archive, and Restore interface. The wizards do not support all of the RMAN commands and options provided by Oracle. A shell script should be written for situations where a template does not provide all of the required functionality.

bpdbsbora   NetBackup for Oracle also provides a utility, `bpdbsbora`, that runs the wizard-generated templates. The utility can also generate a shell script from a NetBackup for Oracle template.

Oracle Terms

Full Backup   A full backup copies all blocks into the backup set, skipping only datafile blocks that have never been used. Note that a full backup is not the same as a whole database backup; full is an indicator that the backup is not incremental.

A full backup has no effect on subsequent incremental backups, which is why it is not considered part of the incremental strategy. In other words, a full backup does not affect which blocks are included in subsequent incremental backups.

Incremental Backup   An incremental backup is a backup of only those blocks that have changed since a previous backup. Oracle allows you to create and restore incremental backups of datafiles, tablespaces, and a database. You can include a control file in an incremental backup set, but the control file is always included in its entirety, no blocks are skipped.

Multi-Level Incremental Backup	Oracle Recovery Manager allows you to create multi-level backups. Each level is denoted by an integer, e.g., 0, 1, 2, etc. A level 0 incremental backup, which is the base of subsequent incremental backups, copies all blocks containing data.

When you generate a level n incremental backup in which n is greater than 0, you back up:

◆ All blocks that have been modified since the most recent backup at a level n or lower (this is the default type of incremental backup, called Differential Incremental Backup)

◆ All blocks that have been modified since the most recent backup at level n-1 or lower (called a Cumulative Incremental Backup)

The benefit of performing multi-level incremental backups is that you do not back up all of the blocks all of the time. Incremental backups at level greater than 0 only copy blocks that were modified, which means that the backup size may be significantly smaller and require much less time. The size of the backup file depends solely upon the number of blocks modified and the incremental backup level.

Differential Incremental Backup	In a differential level n incremental backup, you back up all blocks that have changed since the most recent backup at level n or lower. For example, in a differential level 2 backup, you back up all blocks modified since the last level 2, level 1, or level 0 backup. Incremental backups are differential by default.
Cumulative Incremental Backup	In a cumulative level n incremental backup, you back up all blocks that have changed since the most recent backup at level n-1 or lower. For example, in a cumulative level 2 backup, you back up all blocks changed since the most recent level 1 or level 0 backup.

Cumulative incremental backups reduce the work needed for a restore by ensuring that you only need one incremental backup from any particular level at restore time. Cumulative backups require more space and time than Differential Incremental Backups, however, because they duplicate the work done by previous backups at the same level.

| *proxy copy* | Proxy copy is an extension to the Oracle Media Management API, which gives media management software more control over database backups and restores. With proxy copy feature, instead of simply writing to or reading from the backup devices, the media management software will manage the entire data transfer. |

Oracle RMAN Terms

Oracle Enterprise Manager	Backup Manager, the Oracle Enterprise Backup component, is a graphical user interface for the Recovery Manager (RMAN). This interface allows you to use the point and click method to perform backups and recoveries. For more information about the Oracle Enterprise Manager, see the *Oracle Enterprise Manager Administrator's Guide*.
Oracle Recovery Manager	Oracle Recovery Manager (RMAN) is used to back up, restore, and recover database files. Oracle Recovery Manager starts Oracle server processes on the target database. These Oracle server processes actually perform the backup and restore. Oracle Recovery Manager performs important backup and recovery procedures, and greatly simplifies the tasks administrators perform during these processes. However, it cannot directly manage the storage devices and media used in its backups and must be integrated with an application that has these capabilities. NetBackup for Oracle on UNIX provides device and media management capability by integrating the Oracle Recovery Manager with NetBackup and its media management software. Other advantages are access to NetBackup's automatic scheduling facilities and the graphical interfaces.
RMAN Repository	An RMAN recovery catalog or the database control file is a repository for information that is used and maintained by Oracle Recovery Manager. Oracle Recovery Manager uses this information to determine how to execute requested backup and restore actions.
	Oracle recommends that you use Oracle Recovery Manager with a recovery catalog, especially if you have 20 (or more) datafiles. However, you are not required to maintain a recovery catalog with Oracle Recovery Manager.

Because most information in the recovery catalog is also available in the target database's control file, Oracle Recovery Manager supports an operational mode where it uses the target database control file instead of a recovery catalog. This operational mode is appropriate for small databases where installation and administration of another database, for the sole purpose of maintaining the recovery catalog, would be burdensome.

For a detailed description of the recovery catalog, refer to the *Oracle Recovery Manager User's Guide and Reference*.

rman command
The rman command is used to start a backup or restore by the Oracle Recovery Manager. Recovery Manager is an integral part of Oracle. This command is described in the *Oracle Recovery Manager User's Guide and Reference*.

RMAN script
The RMAN script specifies the commands that Oracle Recovery Manager will perform (for example, backups, and restores). This command file is described in the *Oracle Recovery Manager User's Guide and Reference* (available from the Oracle Corporation).

The *Oracle Recovery Manager User's Guide and Reference* explains the rman commands and the command script files. Refer to that guide for details on command syntax and improving performance.

There are example RMAN scripts in the directory named *install_path*/netbackup/ext/dbext/oracle/samples/rman/. These are shell scripts that execute RMAN commands and are fully commented to explain the features used. We recommend that you review these examples. You can use them as a starting point for developing backup, restore, and recovery scripts.

Technical Overview of NetBackup for Oracle on UNIX

The example network below shows the major components in a NetBackup for Oracle ServerFree Agent configuration.

The storage devices are connected to the NetBackup master server. A NetBackup media server can access the storage devices through the master server. Both the master server and the media server must have NetBackup server software installed.

The host with the database must be a NetBackup client and have NetBackup for Oracle ServerFree Agent installed.

Oracle Recovery Manager

Host 2
                    Host 3

Oracle Recovery Manager

DB

Recovery Catalog

DB

Oracle instance to backup

Oracle Recovery Manager

NetBackup for Oracle

NetBackup Client

Network (TCP/IP)

NetBackup Server

Host 1*

*A UNIX operating system

DB

NetBackup database

Tape or Optical
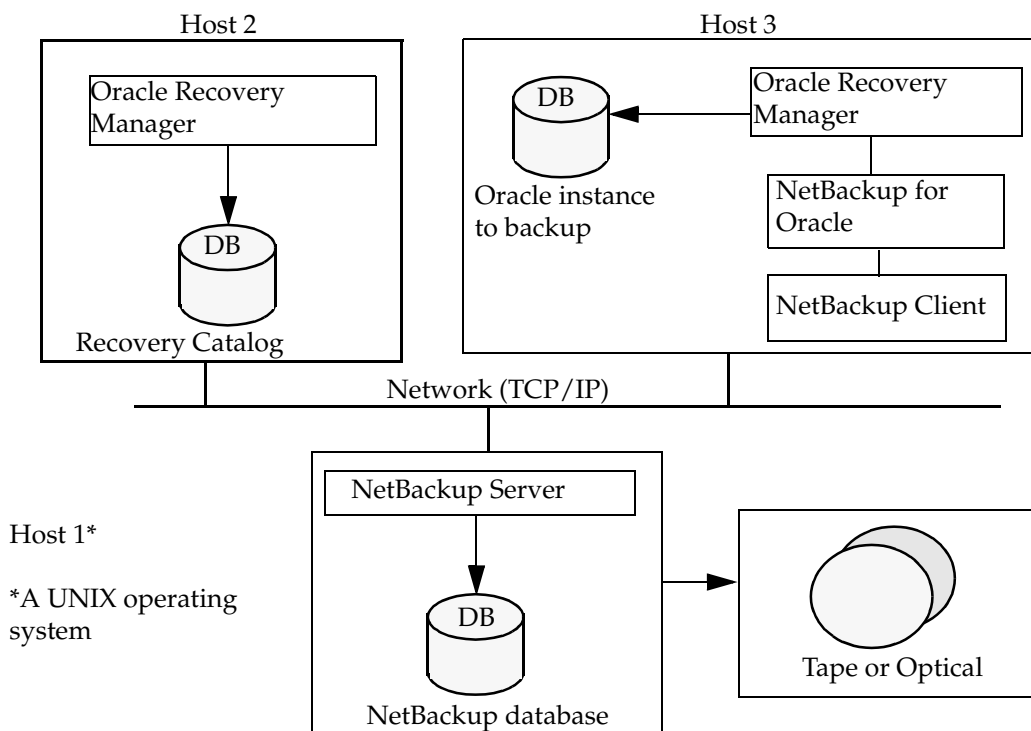
How Does the Oracle Recovery Manager Work?

The Oracle Recovery Manager (RMAN) performs a wide variety of automated backup and recovery functions. During backup or restore, RMAN provides the interface to the databases and performs the actual extraction and insertion of data.

To start a database backup or restore, the database administrator must execute a command called rman. This command can be executed from the command line, a script, or an application such as NetBackup. The RMAN script is used as a parameter to the rman command and defines the commands to be performed (for example, backup or restore). The RMAN script also defines other components of the operation such as the database objects to be backed up or restored.

During a backup or restore, the Oracle Recovery Manager controls the data streams going into or out of a database. The Oracle Recovery Manager can access storage devices when it is integrated with a media management system, such as that provided by NetBackup and its Media Manager.

For more information, read the *Oracle Recovery Manager User's Guide and Reference*.

How Does NetBackup for Oracle on UNIX Work?

Following are prerequisites for performing backups to tape:

◆ Linking with NetBackup and its Media Manager

◆ Generating unique file names

NetBackup for Oracle on UNIX includes a library of functions that enable the Oracle Recovery Manager to use NetBackup and its Media Manager. To link with this library, see "Linking Oracle8 with NetBackup for Oracle on UNIX" on page 28.

When using the backup command, each resulting backup set contains at least one backup piece (datafile, datafile copy, control file, or archive log) from the target database. You must give each backup piece a unique name using the format operand. Several substitution variables are available to aid in generating unique names. You can specify the format operand in the backup command or in the allocate channel command. NetBackup considers the backup piece name as the file being backed up, so this name must be unique in the catalog. In Oracle 8i or later, the RMAN backup will fail with a file already exists in catalog error.

Note For performance reasons, NetBackup for Oracle on UNIX strongly recommends that an RMAN timestamp be placed at the end of each backup piece name. Without this timestamp, there will be performance degradation as the NetBackup catalog grows. To guarantee uniqueness and optimal performance, specify each backup file name as:

*tag*_%s_%p_%t for Oracle 8.0.x
*tag*_%U%_%t for Oracle8i and later

where
tag is any user-defined string
    %s specifies the backup set number

%p specifies the backup piece number with the backup set

%t specifies the backup set timestamp

%U specifies a shorthand for %u_%p_%c that guarantees uniqueness in generated file names.

For more information on generating unique file names, refer to the *Oracle Recovery Manager User's Guide and Reference*.

NetBackup users or schedules start database backups by specifying a NetBackup for Oracle RMAN script in the file list of the Oracle policy. The NetBackup for Oracle RMAN script specifies the backup commands that Oracle Recovery Manager will perform on the client.

For a backup:

1. The rman command starts the requested operation on the databases.

2. When the process requires media to store backup data, RMAN starts a user-directed backup by issuing a backup request.

3. The NetBackup master server connects to NetBackup for Oracle on UNIX on the client and transfers the database data to secondary storage.

A restore works in essentially the same manner except that RMAN issues a restore request. This causes NetBackup to retrieve the data from secondary storage and send it to NetBackup for Oracle on UNIX on the client.

Since the Oracle Recovery Manager supports parallel operations, a single rman execution can start more than one backup or restore on the NetBackup system.

The status for an RMAN operation is stored in the Oracle Recovery Manager catalog or in the database control file. This is the same status that is indicated by the output of the script used to run the backup or restore. This is the only status that a database administrator must check to verify that a backup or restore has been successful.

NetBackup also logs status, but only for its own part of the operation. The database administrator cannot use the NetBackup status to determine whether rman was successful. Errors can occur in rman that do not affect NetBackup and are not recorded in its logs.

Incremental Backup of Oracle Databases

The Oracle Recovery Manager provides true incremental backups. An incremental backup will back up datafiles, which include only the blocks that have been changed since the last incremental.

Refer to the *Oracle Server Administrator's Guide* and the *Oracle Recovery Manager User's Guide and Reference* for more complete details on the backup and recovery process.

Proxy Copy and NetBackup for Oracle ServerFree Agent

This section describes how proxy copy works with the NetBackup for Oracle ServerFree Agent.

Proxy Copy

Proxy copy is an extension to Oracle's Media Management API. It gives NetBackup for Oracle ServerFree Agent more control over database backups and restores. Instead of simply writing to or reading from the backup devices, NetBackup for Oracle ServerFree Agent manages the entire data transfer between the disks containing Oracle datafiles and Netbackup and its Media Manager.

Note Currently, Oracle supports proxy backups for datafiles only. Backups of control files and archive logs must be performed using RMAN non-proxy backups.

With proxy copy, RMAN provides a list of files, requiring backup or recovery, to NetBackup for Oracle ServerFree Agent. NetBackup for Oracle ServerFree Agent makes all decisions regarding how the data is moved and when to move the data.

Backups and restores remain tightly integrated with RMAN and its catalog, greatly simplifying administration tasks.

How does NetBackup for Oracle ServerFree Agent Work?

NetBackup for Oracle ServerFree Agent extends the functionality of NetBackup for Oracle on UNIX. It integrates the Oracle proxy copy feature with the frozen image services and offhost backup technology of NetBackup ServerFree Agent. It coordinates the backup and restore of Oracle datafiles using NetBackup ServerFree Agent (and NetBackup for EMC Fastrax Agent for datafiles on EMC Symmetrix disk arrays).

Note To use NetBackup for Oracle ServerFree Agent you must first have a thorough understanding of NetBackup 4.5 ServerFree Agent. Refer to *NetBackup 4.5 ServerFree Agent System Administrator's Guide* or *NetBackup 4.5 for EMC Fastrax System Administrator's Guide.*

Also, review "How Does NetBackup for Oracle on UNIX Work?" on page 10 before proceeding with this section.

NetBackup users or schedules start database backups or restores by specifying a NetBackup for Oracle RMAN script in the file list of the Oracle policy. The NetBackup for Oracle RMAN script specifies backup commands that the Oracle Recovery Manager will perform on the client.

With proxy copy, RMAN provides a list of files, requiring backup or recovery, to NetBackup for Oracle ServerFree Agent. NetBackup for Oracle ServerFree Agent uses the NetBackup ServerFree Agent interface to perform the actual data movement.

When Oracle performs proxy copy backups, it puts the datafile being backed up into backup mode. Then NetBackup ServerFree Agent creates a frozen image of the data. Once the frozen image has been created, NetBackup for Oracle ServerFree Agent signals back to Oracle to take the datafiles out of backup mode. Therefore the datafiles that are being backed up will be in backup mode only for the period of time necessary to capture a frozen image of the data.

At the beginning of the backup, NetBackup for Oracle ServerFree Agent groups all files it received from RMAN based on the file system they reside on. It then starts a backup for all files on each file system at the same time. The number of proxy copy backup streams to be started on each file system can be controlled by the NB_ORA_PC_STREAMS variable, specified with the send command in the RMAN script. However, the number of parallel streams started will never be larger than the number of files on the file system. Each raw device can be used for only one Oracle database datafile, so the agent always starts only one backup stream for a raw partition or volume. If NB_ORA_PC_STREAMS is not specified, by default the agent starts one backup per file system.

The number of file systems, raw partitions, or logical volumes to be backed up concurrently can be controlled by the NB_ORA_PC_FILESYSTEMS variable, specified with the send command in the RMAN script. If this variable is not specified, by default the agent initiates backups of all target file systems and raw devices concurrently. For more information on how to use these variables, refer to "Environment Variables Set Up by a User" on page 63.

To make sure that NetBackup for Oracle ServerFree Agent uses the proper policy and schedule, use RMAN send command to pass the NB_ORA_POLICY and NB_ORA_PC_SCHED variables. Refer to "Example 4, hot_database_backup_proxy.sh" on page 140.

To configure a policy for Oracle ServerFree backups a user needs to configure:

◆ Frozen image methods for file systems (raw partitions or logical volumes) where the datafiles reside

◆ A backup method

◆ Automatic Full Backup schedule type to perform proxy copy frozen image and offhost backups of the datafiles

◆ Application Backup schedule type to back up a control file and archive logs. Note that Oracle does not support proxy backups of database controlfiles and archived redo logs. These files are backed up using standard RMAN operation.

For detailed instructions on configuring an Oracle policy refer to "Configuring a NetBackup Policy" on page 45.

Proxy copy backups can be used as a part of the incremental strategy with non-proxy RMAN backups. Oracle allows you to create proxy copy backup level 0, which can be the base for subsequent non-proxy incremental backups (level 0 or greater).

Quick I/O Oracle Datafiles

NetBackup for Oracle ServerFree Agent fully supports backups and restores of the Quick I/O Oracle datafiles. A Quick I/O file consists of two components: a hidden file with a space allocated for it, and a link that points to the Quick I/O interface of the hidden file. On the backup, NetBackup for Oracle ServerFree Agent follows the symbolic link and backs up both components of the Quick I/O file: the symbolic link and the hidden file.

During the restore job, NetBackup for Oracle ServerFree Agent will restore both components from the backup image. If either one or both of the components is missing, NetBackup for Oracle ServerFree Agent will create the missing component(s).

Script Examples for Proxy Copy Backups

Refer to the following NetBackup for Oracle ServerFree Agent scripts for proxy database backup.

◆ "Example 4, `hot_database_backup_proxy.sh`" on page 140

◆ "Example 5, `hot_tablespace_backup_proxy.sh`" on page 146

Sequence of Operation

The backup sequence of operation is:

1. NetBackup for Oracle ServerFree Agent receives a list of datafiles to back up.

Each datafile is identified by a unique backup file name in the NetBackup catalog. To insure this, you must give each datafile a unique name using the `format` operand. You can use the same technique as with non-proxy backups.

Note For performance reasons, NetBackup for Oracle on UNIX strongly recommends that an RMAN timestamp be placed at the end of each backup piece name. Without this timestamp, there will be performance degradation as the NetBackup catalog grows. To guarantee uniqueness and optimal performance, specify each backup file name as:

<tag>`_%U%_%t`

where
<tag> is any user-defined string

`%t` specifies the backup set timestamp

`%U` specifies a shorthand for `%u_%p_%c` that guarantees uniqueness in generated file names.

2. NetBackup for Oracle ServerFree Agent groups the datafiles based on the file systems they belong to.

3. NetBackup for Oracle ServerFree Agent uses the value of the NB_ORA_PC_FILESYSTEMS variable to determine how many file systems (raw devices) to back up concurrently.

4. NetBackup for Oracle ServerFree Agent uses the value of the NB_ORA_PC_STREAMS variable to determine the number of backup streams to start for each file system.

Any of the datafiles proxy backed up in one session may be restored individually later, or they may be restored in the same proxy copy session along with other backup files created in different proxy copy sessions.

The restore sequence of operation is:

1. NetBackup for Oracle ServerFree Agent receives a list of datafiles to restore.

2. NetBackup for Oracle ServerFree Agent sends a restore request to the NetBackup server for all files in the list.

3. The NetBackup for Oracle ServerFree Agent waits for all files in the file list to be restored by NetBackup.

Installation 2

This chapter describes the NetBackup for Oracle ServerFree Agent installation procedure. It includes a section on installation prerequisites. It also contains a section on configuring Oracle Recovery Manager.

Installation Prerequisites

Before installing NetBackup for Oracle ServerFree Agent, be sure you have completed the following procedures:

❖ Install Oracle.

❖ Install NetBackup 4.5 ServerFree Agent

❖ Install NetBackup 4.5 for EMC Fastrax to back up datafiles on an EMC Symmetrix disk array.

Now you are ready to install NetBackup for Oracle ServerFree Agent on the client where you will be backing up the databases. Refer to the next section for detailed instructions on installing NetBackup for Oracle ServerFree Agent.

Note For HP and Solaris clients, the NetBackup for Oracle installation package includes three different NetBackup for Oracle database agents. This combination is for installation purposes only. Each NetBackup for Oracle database agent will continue to function with no dependencies on the other agents. See the release notes for further explanation.

Installing NetBackup for Oracle ServerFree Agent

There are two ways to install database agent software.

◆ Remote Installation - Loads the software on a master server. The user will then push the database software out to affected clients.

Refer to the following section.

◆ Local Installation - Loads and installs the software only to the local machine.

Refer to "Local Installation of NetBackup for Oracle ServerFree Agent" on page 26.

Note During the installation , you will be prompted to enter a UNIX database administrators' group name. This is the name of the group that the Oracle software owner belongs to. In most cases the name is **dba**. If you have multiple Oracle installations on your system, with different software owner names, make sure that they all belong to a single UNIX database administrators' group.

Remote Installation of NetBackup for Oracle ServerFree Agent

During a remote installation, NetBackup for Oracle ServerFree Agent files are loaded onto the current machine, which must be a UNIX master server. The software will then be distributed to the clients and installed.

Installation Requirements

◆ The version of the NetBackup Client and the version of NetBackup for Oracle ServerFree Agent must be the same (e.g., 4.5).

◆ There must be adequate disk space on each client that will receive the software.

Less than one megabyte of additional disk space is required in the client's *install_path*/netbackup directory. However, more disk space may be necessary at run time.

◆ NetBackup version 4.5 software is installed and operational on each NetBackup for Oracle ServerFree Agent client.

This also means that the directory *install_path*/netbackup already exists on each NetBackup for Oracle ServerFree Agent client.

Remote Install Procedure

1. Log in as the root user on the server.

   If you are already logged in, but are not the root user, execute the following command.

   su - root

2. Make sure a valid license key for NetBackup for Oracle ServerFree Agent has been registered. To view or add license keys, perform one of the following:

   - Run the command
     install_path/netbackup/bin/admincmd/get_license_key.

   - Open the NetBackup Administration Console and from the **Help** menu choose **License Keys**.

3. Insert the CD-ROM into the drive.

4. Change the working directory to the CD-ROM directory.

   cd /CD_mount_point

5. Load the software on the server by executing the install script.

   ./install

   a. Select the NetBackup Database Agent Software option.

      The following prompt will appear:

      Do you want to do a local installation? (y/n) [n]

   b. Answer **n**.

      You are presented with a menu of all database agents available on the CD-ROM.

   c. Select the NetBackup for Oracle+ option.

   d. Enter **q** to quit selecting options.

      A prompt will appear asking if the list is correct.

   e. Answer **y**.

      The install script identifies the types of client software loaded during the installation of the NetBackup server. By default, any matching NetBackup for Oracle ServerFree Agent software will automatically be loaded. If there are more platforms available, the script displays a menu giving you the opportunity to add

more client types to the default list. Once the list is complete, database agent version files, compressed tar files and the install_dbext script are copied to directory *install_path*/netbackup/dbext.

6. Distribute and install the NetBackup for Oracle ServerFree Agent software on each client as described in the next sections.

There are two types of installs.

- *initial install* — Use an initial install if the clients you intend to update have not been configured into policies of type Oracle.

- *upgrade install* — Use an upgrade install if all the clients you intend to update already have been configured into policies of type Oracle.

Note The NetBackup version level (for example, 4.5) running on the clients you wish to update must be the same as the version level of the NetBackup for Oracle ServerFree Agent being installed.

Initial Install Procedure

1. Execute the following command to create a file containing a list of clients currently configured in the NetBackup database.

```
cd install_path/netbackup/bin
./admincmd/bpplclients -allunique -noheader > filename
```

where *filename* is the name of the file to contain the list of unique clients. If no clients have been configured in the NetBackup database, and therefore *filename* is empty, create *filename* using the same format as that generated by bpplclients.

bpplclients generates output in following format:

hardware operating_system client_name

where:

hardware is the hardware name. For example, execute the ls command in directory *install_path*/netbackup/client.

operating_system is the operating system name. For example, execute the ls command in directory *install_path*/netbackup/client/*hardware*.

client_name is the name of the client.

For example, the contents of *filename* might look like this:

```
Solaris Solaris2.6 curry
HP9000-800 HP-UX11.00 blaze
```

2. Edit *filename*.

This is an optional step. Use it if the contents of *filename* need to be changed. Edit *filename* to contain only those clients you wish to update with NetBackup for Oracle ServerFree Agent software.

3. Specify *filename* on the `update_dbclients` command.

For example:

```
cd install_path/netbackup/bin
./update_dbclients Oracle -ClientList filename
```

Only clients listed in *filename* will be updated.

Note Because all of the NetBackup for Oracle database agents are included in the installation package, one remote installation procedure will install all of the NetBackup for Oracle agents licensed to each client.

Upgrade Install Procedure

1. Execute the following command.

```
cd install_path/netbackup/bin
./update_dbclients Oracle ALL ALL
```

This command will look at all possible clients and only update the ones currently in the Oracle policy type.

Instead of `ALL ALL`, you may use `-ClientList` *filename* as explained in "Initial Install Procedure" on page 21.

Note With an initial or upgrade install, some clients may be skipped and not updated. Possible reasons are:
- the client is a PC client (which cannot be updated from a UNIX server)
- NetBackup for Oracle ServerFree Agent does not support that client's platform type
- the NetBackup for Oracle ServerFree Agent software for that client type was not loaded onto the server in step 5 of the "Remote Install Procedure" on page 20
- (if using the `ALL ALL` method) the client does not belong to the Oracle policy type
All skipped clients are available in a file whose name is displayed by `update_dbclients`.

The number of updates required to distribute the software to the clients is displayed.

If more than one update will occur, you will see the following prompt:

```
Enter the number of simultaneous updates you wish to take place. 1 - max dflt
```

(must be in the range of 1 to *max*; default: *dflt*).

where:

> *max* is the maximum number of simultaneous updates that is allowed. The value displayed will be a number ranging from 1 to 30.

> *dflt* is the default number the program will use if you press **Enter**. The value displayed will be a number ranging from 1 to 15.

For example, if three client updates will be performed, the *max* and *dflt* values shown would be 3. If 50 client updates will be performed, the *max* value shown would be 30 and the *dflt* value shown would be 15.

`update_dbclients` will start the number of updates that you specify. If this number is less than the total number of client updates to be performed, new updates will start as the previous updates finish until all of the updates have been completed.

Based on your answer, the time it will take to update the clients is displayed, followed by this question:

```
Do you want to upgrade the clients now? (y/n) [y]
```

2. Enter **y** or **n** for the prompt.

   If you answer **n**, `update_dbclients` will quit and leave the list of clients it would have updated in a file. This file can later be used by the `-ClientList` parameter mentioned previously.

   Answer **y** to continue the installation process.

   - If the configuration information for a client is unknown, you will be prompted for it. For example:

   ```
   -----> Client curry

   Please specify the UNIX database administrators' group for
   the Oracle software owner:
   ```

   > Type the UNIX group to which the Oracle software owner belongs. In most cases, the group entered will be **dba**. You will have an opportunity to validate your answer.

   - If the `update_dbclients` command is able to determine a client's configuration, it will not prompt.

   To change a client's Oracle software owner group configuration information later, you must log onto the client and execute the command *install_path*`/netbackup/bin/install_oracle`.

- If more than one client is being updated, the configuration information for the previous client (whether determined by asking explicitly or by `update_dbclients` itself) establishes a default answer for the next client. This makes configuration simpler when the clients have been similarly configured. Therefore, you may see a display similar to the following:

```
-----> Client guava

Please specify the UNIX database administrators' group for
the Oracle software owner:

The previous response to this question was:
dba

Use the previous response? (y/n) [y]
```

Example: Let's assume that you want to update the following three clients.

curry

guava

hat

As far as the script is concerned, the configuration information for all three clients is unknown.

Assume the Oracle software owner belongs to group **dba** on curry, so when you are prompted for configuration information for curry, specify **dba** as a valid group for the Oracle software owner.

The next client is guava.

- If client guava also has **dba** as its Oracle software owner group, the answer to the `Use the previous response?` prompt will be **y**.

- If client guava has a different configuration, the answer to the `Use the previous response?` prompt will be **n**. The `Please specify the UNIX database administrators' group for the Oracle software owner:` prompt will re-display. At this point you specify a different group.

The next client is hat . The Oracle software owner group you entered for guava becomes the "previous response" displayed for hat.

If the `update_dbclients` command was successful in distributing the software to the client, it will automatically run the `install_dbext` script on the client. Configuration information determined in step 2 on page 23 is used to complete the installation. If `install_dbext` has successfully completed, there will be a version file in directory *install_path*/netbackup/ext that contains the version of NetBackup for Oracle ServerFree Agent that was installed and an installation timestamp. The `update_dbclients` command displays a note on whether the update was successful for

each client. When the `update_dbclients` command has completed, it displays a file name that contains a complete log of what happened for each client. If the update failed for any client, the log file should be examined to determine the problem.

Local Installation of NetBackup for Oracle ServerFree Agent

During a local installation, the NetBackup for Oracle ServerFree Agent files are extracted and installed. You also are prompted for configuration information. The local machine can be a client or a master server that also happens to be a client.

Installation Requirements

◆ The version of the NetBackup Client and the version of NetBackup for Oracle ServerFree Agent are the same (e.g., 4.5).

◆ The local machine must have adequate disk space.

Less than one megabyte of additional disk space is required in the *install_path*/netbackup directory. However, more disk space may be necessary at run time.

◆ NetBackup version 4.5 client software is installed and operational.

This also means that the *install_path*/netbackup directory already exists.

Local Install Procedure

1. Log in as the root user on the machine.

If you are already logged in, but are not the root user, execute the following command.

   su - root

   - If the local machine is a client, go to step 3.

   - If the local machine is a server, go to step 2.

2. Make sure a valid license key for NetBackup for Oracle ServerFree Agent has been registered. To view or add license keys, perform one of the following:

   - Run the command
     install_path/netbackup/bin/admincmd/get_license_key.

   - Open the NetBackup Administration Console and from the **Help** menu choose **License Keys**.

3. Insert the CD-ROM into the drive.

4. Change the working directory to the CD-ROM directory.

   cd /CD_mount_point

5. Load and install the software by executing the install script.

> **Note** It is expected that the NetBackup version level (for example, 4.5) running on the local machine matches the version level of the database agent being installed.

`./install`

a. Select the NetBackup Database Agent Software option.

The following prompt will appear:

```
Do you want to do a local installation? (y/n) [n]
```

b. Answer **y**.

You are presented with a menu of all database agents available on the CD-ROM.

c. Select the NetBackup for Oracle+ option.

d. Enter **q** to quit selecting options.

A prompt will appear asking if the list is correct.

e. Answer **y**.

The following actions will occur:

- The version file, compressed tar file and install_dbext script will be loaded to directory *install_path*/netbackup/ext/dbext.

- The install script will automatically execute the install_dbext script.

- After the install_dbext script unbundles the compressed tar file containing the NetBackup for Oracle ServerFree Agent software, you will be prompted for the following configuration information.

  ```
  Please specify the UNIX database administrators' group for
  the Oracle software owner:
  ```

  Type the UNIX group to which the Oracle software owner belongs. In most cases, the group entered will be **dba**.

- If install_dbext has successfully completed, there will be a version file in directory *install_path*/netbackup/ext/ that contains the version of NetBackup for Oracle ServerFree Agent that was installed and an installation timestamp.

Oracle Recovery Manager

To configure the Oracle8 Recovery Manager, refer to the Oracle8 Server Backup and Recovery Guide.

Linking RMAN with a library provided by NetBackup for Oracle on UNIX is the only NetBackup requirement. Before writing to sequential devices such as tape, you must link the Oracle8 Server software with the media management API library installed by NetBackup for Oracle on UNIX. Oracle uses this library when it needs to write to, or read from devices supported by Media Manager.

The default location for the NetBackup API library is /*install_path*/netbackup/bin, while its name is:

◆ `libobk.so.1` on Solaris (32-bit)

◆ `libobk.so64.1` on Solaris (64-bit)

◆ `libobk.sl` on HP-UX (32-bit)

◆ `libobk.sl64` on HP-UX (64-bit)

Refer to Chapter 5, "Troubleshooting," if you encounter problems.

Linking Oracle8 with NetBackup for Oracle on UNIX

This section documents the preferred way to link Oracle with the NetBackup API library. The linking process is not consistent across hardware platforms and Oracle Server release levels and on some platforms, it is not consistent across OS levels. For this writing, it was not possible to include all of the supported platform, OS level, and Oracle Server level combinations. However, OS levels are specified when known differences occur.

Runtime problems may occur if these linking instructions are not followed.

▼ **To link Oracle8 with NetBackup for Oracle on UNIX**

1. Become the Oracle user.

```
su - oracle
```

2. Shutdown all of the Oracle instances (SIDs). The following example is for Oracle8i.

```
svrmgrl
connect internal
shutdown
exit
```

Change the ORACLE_SID environment variable to any other SID that uses this Oracle installation and repeat above commands.

3. Set up automatic or manual linking.

- **-** Automatic Linking

  Run the `oracle_link` script located in

  install_path`/netbackup/bin/`

  This script determines the Oracle version level and then links Oracle to NetBackup accordingly. All output from this script will be captured in a `/tmp/make_trace.pid` file. To change the trace file location, change the *MAKE_TRACE* variable in the `oracle_link` script.

- **-** Manual Linking

  Make a copy of the existing Oracle executable.

  ```
  cd ${ORACLE_HOME}/bin
  cp oracle oracle.orig
  ```

  Starting with version 8.0.4, Oracle is shipped with default media management software. To disable this media management software you must remove or rename the libobk file from *${ORACLE_HOME}*/lib. It may be a file or a symbolic link. To satisfy this and other linking requirements, follow the instructions below for your specific platform and version of Oracle.

❖ On (32-bit or 64-bit) Solaris using 32-bit Oracle software.

    a. Create a symbolic link from Oracle to the NetBackup API library

    Oracle8 Release 8.0.3, 8.0.4, 8.0.5, 8.0.5.1, and 8.0.6 and Oracle8i Release 8.1.5 and 8.1.7

```
cd ${ORACLE_HOME}/lib
ls -l libobk.so
```

    If libobk.so exists.

```
mv libobk.so libobk.so.orig
```

```
ln -s /install_path/netbackup/bin/libobk.so.1 libobk.so
```

    Oracle8i Release 8.1.6

```
cd ${ORACLE_HOME}/lib
ls -l libdsbtsh8.so
```

    If libobk.so exists.

```
mv libdsbtsh8.so libdsbtsh8.so.orig
```

```
ln -s /install_path/netbackup/bin/libobk.so.1 libdsbtsh8.so
```

    Oracle8i Release 8.1.5, 8.1.6, and 8.1.7

    This is the only required step. You no longer have to make a new Oracle executable to use the NetBackup for Oracle API library. With Oracle 8i, the Oracle executable always looks for the shared library: ${ORACLE_HOME}/lib/libobk.so.

    b. Make a new Oracle executable

```
cd ${ORACLE_HOME}/rdbms/lib
```

◆ Procedure to verify library is linked, VERITAS path should be first one.

```
$ ldd  $ORACLE_HOME/bin/oracle
    libobk.so =>        /install_path/netbackup/bin/libobk.so
    libnsl.so.1 =>   /usr/lib/libnsl.so.1
    libsocket.so.1 =>        /usr/lib/libsocket.so.1
    libdl.so.1 =>    /usr/lib/libdl.so.1
    libc.so.1 =>     /usr/lib/libc.so.1
    libaio.so.1 =>   /usr/lib/libaio.so.1
    libm.so.1 =>     /usr/lib/libm.so.1
    libmp.so.2 =>    /usr/lib/libmp.so.2
```

❖ On 64-bit Solaris using 64-bit Oracle software

    a. Create a symbolic link from Oracle to the NetBackup API library

    Oracle8i Release 8.1.6

```
cd ${ORACLE_HOME}/lib64
ls -l libdsbtsh8.so
```

If libobk.so exists.

```
mv libdsbtsh8.so libdsbtsh8.so.orig
```

```
ln -s /install_path/netbackup/bin/libobk.so.1 libdsbtsh8.so
```

Oracle8i Release 8.1.7

```
cd ${ORACLE_HOME}/lib64
ls -l libobk.so
```

If libobk.so exists.

```
mv libobk.so libobk.so.orig
```

```
ln -s /install_path/netbackup/bin/libobk.so64.1 libobk.so
```

Oracle8i Release 8.1.6 and 8.1.7

This is the only required step. You no longer have to make a new Oracle executable to use the NetBackup for Oracle API library. With Oracle8i the Oracle executable always looks for the shared library: ${ORACLE_HOME}/lib/libobk.so.

◆ Procedure to verify library is linked, VERITAS path should be first one.

```
$ ldd  $ORACLE_HOME/bin/oracle
   libobk.so =>        /install_path/netbackup/bin/libobk.so
   libnsl.so.1 =>   /usr/lib/libnsl.so.1
   libsocket.so.1 =>        /usr/lib/libsocket.so.1
   libdl.so.1 =>   /usr/lib/libdl.so.1
   libc.so.1 =>    /usr/lib/libc.so.1
   libaio.so.1 =>   /usr/lib/libaio.so.1
   libm.so.1 =>    /usr/lib/libm.so.1
   libmp.so.2 =>   /usr/lib/libmp.so.2
```

❖ On (32-bit or 64-bit) HP-UX using 32-bit Oracle software.

  a. Set the Oracle user's library search path

  ${ORACLE_HOME}*/lib must be in the Oracle user's library search path to prevent linking and runtime load errors. If *${ORACLE_HOME}/lib is not in the search path, the NetBackup API shared library (libobk) will not be found when linking and a runtime load error will occur when an application, such as svrmgrl, is executed. On HP-UX, make sure that the environment variable, SHLIB_PATH, is properly set and if you are using the C shell, edit the Oracle user's .cshrc login file to include the line:

  ```
  setenv SHLIB_PATH=${ORACLE_HOME}/lib:$SHLIB_PATH
  ```

  Otherwise, edit the Oracle user's .profile login file to include the lines:

  ```
  SHLIB_PATH=${ORACLE_HOME}/lib:$SHLIB_PATH
  export SHLIB_PATH
  ```

  b. Create a symbolic link from Oracle to the NetBackup API library

  ```
  cd ${ORACLE_HOME}/lib
  ls -l libobk.sl
  ```

  If libobk.sl exists.

  ```
  mv libobk.sl libobk.sl.orig
  ```

  ```
  ln -s /install_path/netbackup/bin/libobk.sl libobk.sl
  ```

  Oracle8i Release 8.1.5, 8.1.6, and 8.1.7

  This is the only required step. You no longer have to make a new Oracle executable to use the NetBackup for Oracle API library. The Oracle executable always looks for the shared library: *${ORACLE_HOME}*/lib/libobk.sl.

◆ Procedure to verify library is linked, VERITAS path should be first one.

  ```
  $ chatr $ORACLE_HOME/bin/oracle
    libobk.so =>        /install_path/netbackup/bin/libobk.so
    libnsl.so.1 =>   /usr/lib/libnsl.so.1
    libsocket.so.1 =>        /usr/lib/libsocket.so.1
    libdl.so.1 =>    /usr/lib/libdl.so.1
    libc.so.1 =>     /usr/lib/libc.so.1
    libaio.so.1 =>   /usr/lib/libaio.so.1
    libm.so.1 =>     /usr/lib/libm.so.1
    libmp.so.2 =>    /usr/lib/libmp.so.2
  ```

❖ On 64-bit HP-UX using 64-bit Oracle software.

    a. Create a symbolic link from Oracle to the NetBackup API library

     Oracle 8.0.5.1 and 8.0.6

```
cd ${ORACLE_HOME}/lib
ls -l libobk.*
```

     If libobk.sl exists.

```
mv libobk.sl libobk.sl.orig
```

     If libobk.a exists.

```
mv libobk.a libobk.a.orig
```

```
ln -s /install_path/netbackup/bin/libobk.sl64  libobk.sl
```

     Oracle8i Release 8.1.6 and 8.1.7

```
cd ${ORACLE_HOME}/lib64
ls -l libobk.*
```

     If libobk.sl exists.

```
mv libobk.sl libobk.sl.orig
```

     If libobk.a exists.

```
mv libobk.a libobk.a.orig
```

```
ln -s /install_path/netbackup/bin/libobk.sl64  libobk.sl
```

     This is the only required step. You no longer have to make a new Oracle executable to use the NetBackup for Oracle API library. The Oracle executable always looks for the shared library: ${ORACLE_HOME}/lib/libobk.so.

    b. Make a new Oracle executable

```
cd ${ORACLE_HOME}/rdbms/lib
```

     Oracle8 Release 8.0.5.1

```
make -f ins_rdbms.mk ioracle LLIBOBK=-lobk
```

Install Oracle9i Recovery Manager

To configure the Oracle9i Recovery Manager, refer to the Oracle9i Server Backup and Recovery Guide.

Linking RMAN with a library provided by NetBackup for Oracle on UNIX is the only NetBackup requirement. Before writing to sequential devices such as tape, you must link the Oracle9i Server software with the media management API library installed by NetBackup for Oracle on UNIX. Oracle uses this library when it needs to write to, or read from devices supported by Media Manager.

The default location for the NetBackup API library is /usr/openv/netbackup/bin; while its name is:

◆   libobk.so.1 on Solaris (32-bit)

◆   libobk.so64.1 on Solaris (64-bit)

◆   libobk.sl on HP-UX (32-bit)

◆   libobk.sl64 on HP-UX (64-bit)

Read Chapter 5, "Troubleshooting," if you encounter problems.

Linking Oracle9i with NetBackup for Oracle on UNIX

This section documents the preferred way to link Oracle with the NetBackup API library. The linking process is not consistent across hardware platforms and Oracle Server release levels and on some platforms, it is not consistent across OS levels. For this writing, it was not possible to include all of the supported platform, OS level, and Oracle Server level combinations. However, OS levels are specified when known differences occur.

Runtime problems may occur if these linking instructions are not followed.

▼ **To link Oracle9i with NetBackup for Oracle on UNIX**

1. Become the Oracle user.

   ```
   su - oracle
   ```

2. Shutdown all of the Oracle instances (SIDs).

   ```
   sqlplus
   sys/change_on_install as sysdba        (login/password)
   shutdown
   exit
   ```

   Change then ORACLE_SID environment variable to any other SID that uses this Oracle installation and repeat above commands.

3. Set up automatic or manual linking.

   - Automatic Linking

     Run the `oracle_link` script located in

     install_path`/netbackup/bin/`

     This script determines the Oracle version level and then links Oracle to NetBackup accordingly. All output from this script will be captured in a `/tmp/make_trace.pid` file. To change the trace file location, change the *MAKE_TRACE* variable in the `oracle_link` script.

   - Manual Linking

     Starting with version 8.0.4, Oracle is shipped with default media management software. To disable this media management software you must remove or rename the libobk file from *${ORACLE_HOME}*/lib. It may be a file or a symbolic link. To satisfy this and other linking requirements, follow the instructions below for your specific platform and version of Oracle.

❖ HP-UX 64-bit using 64-bit Oracle software.

a. Create a symbolic link from Oracle to the NetBackup API library

Oracle 9i Release 9.0.1

```
cd ${ORACLE_HOME}/lib
ls -l libobk.*
```

If libobk.sl exists.

```
mv libobk.sl libobk.sl.orig
```

If libobk.a exists.

```
mv libobk.a libobk.a.orig
ln -s /usr/openv/netbackup/bin/libobk.sl64  libobk.sl
```

This is the only required step. You no longer have to make a new Oracle executable to use the NetBackup for Oracle API library. With Oracle 9i the Oracle executable always looks for the shared library:
${ORACLE_HOME}/lib/libobk.so.

❖ Solaris (32-bit or 64-bit) using 32-bit Oracle software.

a. Create a symbolic link from Oracle to the NetBackup API library

```
cd ${ORACLE_HOME}/lib
ls -l libobk.so
```

If libobk.so exists.

```
mv libobk.so libobk.so.orig
ln -s /usr/openv/netbackup/bin/libobk.so.1 libobk.so
```

Oracle9i Release 9.0.1

This is the only required step. You no longer have to make a new Oracle executable to use the NetBackup for Oracle API library. With Oracle 8i the Oracle executable always looks for the shared library:
${ORACLE_HOME}/lib/libobk.so.

❖ Solaris 64-bit using 64-bit Oracle software.

   a. Create a symbolic link from Oracle to the NetBackup API library

```
cd ${ORACLE_HOME}/lib
ls -l libobk.so
```

If libobk.so exists.

```
mv libobk.so libobk.so.orig
ln -s /usr/openv/netbackup/bin/libobk.so64.1 libobk.so
```

Oracle9i Release 9.0.1

This is the only required step. You no longer have to make a new Oracle executable to use the NetBackup for Oracle API library. With Oracle 8i the Oracle executable always looks for the shared library: ${ORACLE_HOME}/lib/libobk.so.

Linking Rollback Procedure

If you run into problems and are unable to re-link Oracle with NetBackup for Oracle ServerFree Agent, you can roll back what you have done by entering the following command:

```
cd $ORACLE_HOME/lib
mv libobk.so.orig libobk.so
```

(or whatever the API library file was originally called).

Install Oracle Recovery Catalog

The recovery catalog is a repository of information that is used and maintained by Oracle Recovery Manager. Oracle Recovery Manager uses the information in the recovery catalog to determine how to execute requested backup and restore actions.

The recovery catalog contains information about:

◆ Datafile and archivelog backup sets and backup pieces

◆ Datafile copies

◆ Archived redo logs and their copies

◆ Tablespaces and datafiles on the target database

◆ Stored scripts, which are named, user-created sequences of RMAN and SQL commands

Oracle recommends you use Oracle Recovery Manager with a recovery catalog, especially if you have 20 (or more) datafiles. However, you are not required to maintain a recovery catalog with Oracle Recovery Manager.

Because most information in the recovery catalog is also available in the target database's control file, Oracle Recovery Manager supports an operational mode where it uses the target database control file instead of a recovery catalog. This mode is appropriate for small databases where installation and administration of another database for the sole purpose of maintaining the recovery catalog would be burdensome.

If a recovery catalog is not used, the following features are not supported:

◆ Tablespace point-in-time recovery

◆ Stored scripts

◆ Restore and recovery when the control file is lost or damaged

Because recovery catalog installation procedures are version specific, refer to your *Oracle Recovery Manager User's Guide and Reference* for details.

Configuration 3

Before attempting to configure NetBackup for Oracle ServerFree Agent, complete the installation procedure as described in the Installation chapter.

The following is the configuration procedure.

1. Configuring the Media Manager

2. Setting the Maximum Jobs per Client Global Attribute

3. Configuring a NetBackup Policy

4. Configuring Run-Time Environment

5. Creating Templates and Shell Scripts

The following sections in this chapter describe each of these steps in detail.

Configuration Using the NetBackup Administration Console

Although the database agent is installed on the NetBackup client, some configuration procedures are performed using the NetBackup Administration Console on the server.

These procedures include:

◆ Configuring the Media Manager

◆ Setting the Maximum Jobs per Client global attribute

◆ Configuring a NetBackup policy

◆ Testing NetBackup for Oracle on UNIX configuration settings

See the next section for instructions on starting the NetBackup Administration Console.

▼ **To launch the NetBackup Administration Console for UNIX**

1. Log onto the UNIX server as root.

2. Start the NetBackup Administration Console by executing:

*install_path***/netbackup/bin/jnbSA &**

The Login dialog box appears.
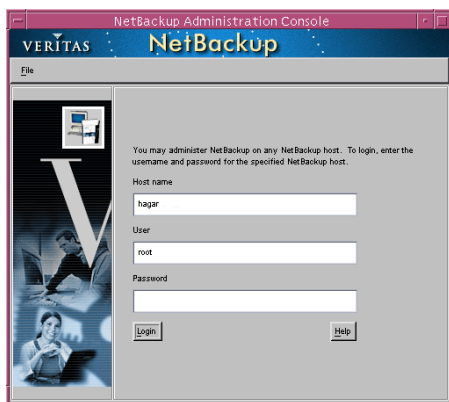
3. Type the name of the master server where you initially want to manage NetBackup. You can specify any NetBackup master server. Indicate the User and Password.

4. Click **Login**. The NetBackup Administration Console appears.
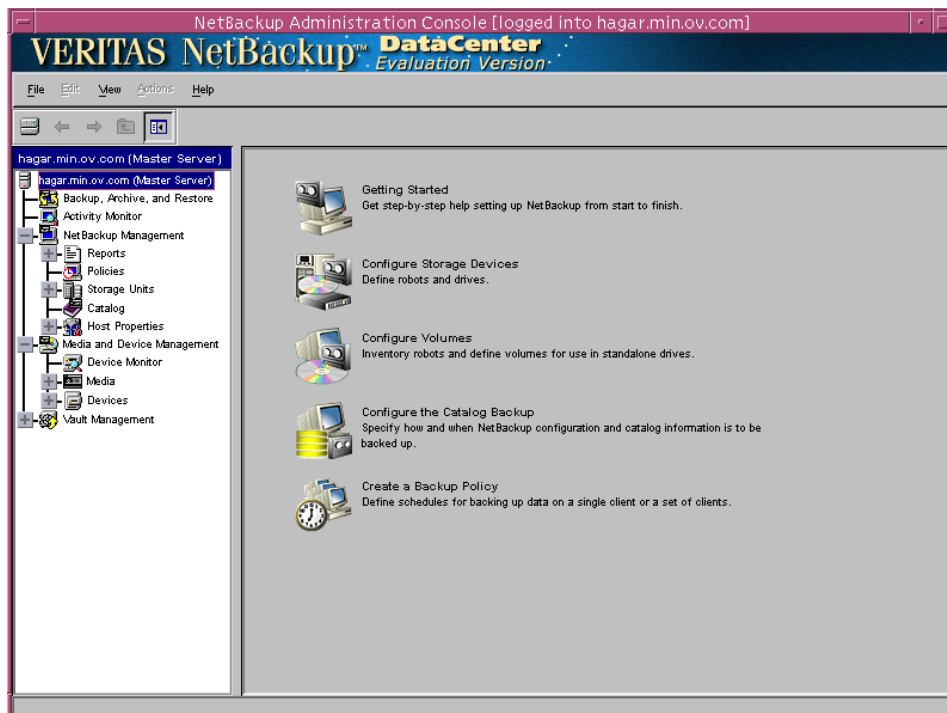
Configuring the Media Manager

Use the Media Manager to configure tapes or other storage units for a NetBackup for Oracle on UNIX configuration.

◆ Refer to the *Media Manager for NetBackup System Administrator's Guide for UNIX* if the NetBackup server is UNIX.

◆ Refer to the *Media Manager for NetBackup System Administrator's Guide for Windows* if the NetBackup server is Windows.

The number of volumes required will depend on the devices used, the size of the Oracle databases that you are backing up, and the frequency of backups.

Setting the Maximum Jobs per Client Global Attribute

The **Maximum jobs per client** attribute value is figured with the following formula.

Max Jobs per Client = *Number of Streams* x *Number of Policies*

Where:

◆ *Number of Policies* is the number of policies that may back up this client at the same time. This number can be greater than one. For example, a client may be in two policies in order to back up two different databases. These backup windows may overlap.

For Oracle non-proxy backups and restores, the number of jobs is hard to determine, as Oracle internally determines when and how many streams to run in parallel to optimize performance. For Oracle proxy backups, the number of streams is determined by two variables: NB_ORA_PC_FILESYSTEMS and NB_ORA_PC_STREAMS. Refer to "Environment Variables Set Up by a User" on page 78 for details.

To avoid any problems, we recommend that you enter a value of 99 for the **Maximum jobs per client** global attribute.

▼ **To set the Maximum jobs per client attribute on a UNIX server**

Use this procedure to set the **Maximum jobs per client** global attribute using the NetBackup Administration Console - Java interface on a Java-capable platform.

1. In the left pane of the NetBackup Administration Console, expand **Host Properties**. Select **Master Servers**.

2. In the right pane, double-click on the server icon. Click **Global Attributes**.

   The Master Server Properties dialog box appears.
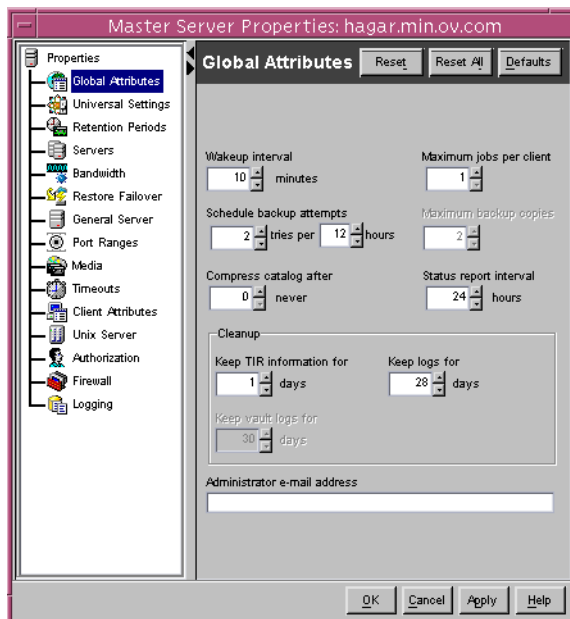
The default value is 1 for **Maximum jobs per client**.

3. Change the **Maximum jobs per client** value to a value equal to the maximum number of backups allowed per client.

Tip	To avoid any problems, we recommend that you enter a value of 99 for the **Maximum jobs per client** global attribute.

Configuring a NetBackup Policy

A NetBackup policy defines the backup criteria for a specific group of one or more clients. These criteria include:

◆ storage unit and media to use

◆ backup schedules

◆ backup templates or script files to be executed on the clients

◆ clients to be backed up

To use NetBackup for Oracle on UNIX, at least one Oracle policy with the appropriate schedules needs to be defined. A configuration can have a single policy that includes all clients or there can be many policies, some of which include only one client.

Most requirements for Oracle policies are the same as for file system backups. In addition to the attributes described here, there are other attributes for a policy to consider. Refer to the *NetBackup System Administrator's Guide* for detailed configuration instructions and information on all the attributes available.

Use this procedure when configuring a policy from a UNIX server.

▼ **To add a new policy**

1. Log onto the server as root.

2. Start the NetBackup Administration Console.

3. If your site has more than one master server, choose the one to which you want to add the policy.

4. In the left pane, click on **Policies**. The right pane splits into a All Policies pane and a details pane.

5. In the All Policies pane, right-click on the Master Server, and click **New**.

The Add a New Policy dialog box appears.
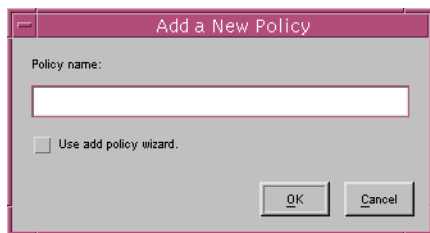
a. In the **Policy name** box, type a unique name for the new policy.

b. Choose whether to use the wizard for configuring the policy. The wizard guides you through the setup process and simplifies it by automatically choosing default values that are good for most configurations. If necessary, you can change the defaults later by editing the policy.

- To use the wizard, select the **Use add policy wizard** box and click **OK**. The wizard starts and you create the policy by following the prompts. When prompted, select the Oracle policy type.

- If you require more control over the settings than the wizard provides, do not select the **Use add policy wizard box** and proceed to step 6.

6. Click **OK**.

A dialog box appears in which you can specify the general attributes for the policy.
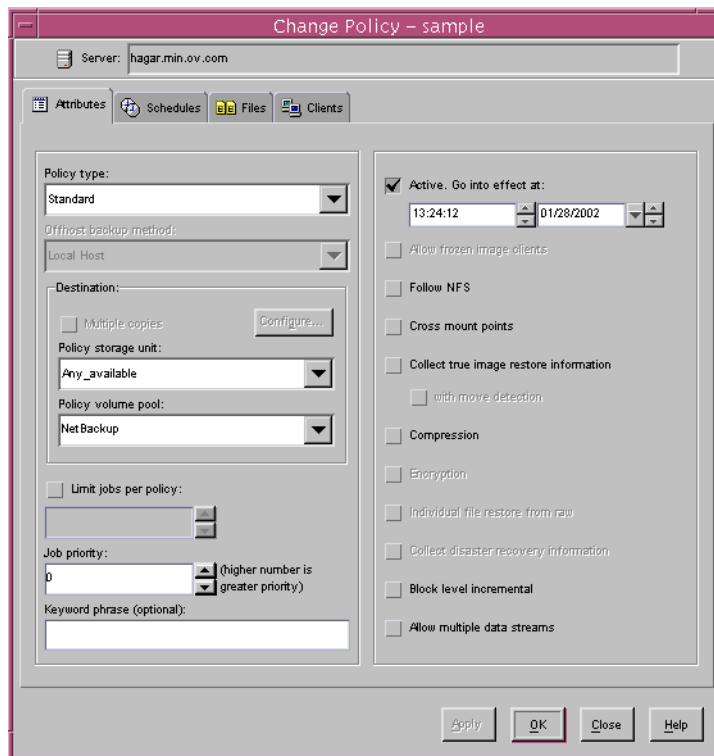
7. From the **Policy type** box, select the Oracle policy type.

The Change Policy dialog box will change as follows.

Available only if
NetBackup ServerFree
Agent software has
been installed.

If you choose Third- Party
Copy Device for Offhost
backup method, select a
particular policy storage
unit (do NOT choose
"Any_available").

Change Policy – TE_Database

Server: sundrops

Attributes | Schedules | Files | Clients

Policy type:
Oracle

Offhost backup method:
Local Host

Destination:
☐ Multiple copies   Configure...

Policy storage unit:
sundrops-dlt-1

Policy volume pool:
NetBackup

☐ Limit jobs per policy:

Job priority:
0   (higher number is
greater priority)

Keyword phrase (optional):

☑ Active. Go into effect at:
04:11:39   10/25/2001

☑ Allow frozen image clients

☐ Follow NFS

☐ Cross mount points

☐ Collect true image restore information

☐ with move detection

☐ Compression

☐ Encryption

☐ Individual file restore from raw

☐ Collect disaster recovery information

☐ Block level incremental

☐ Allow multiple data streams

Apply | OK | Close | Help

If you choose Third-Party Copy Device or NetBackup Media
Server, the Allow frozen image clients checkbox is
pre-selected and greyed out.

8. Select the **Allow frozen image clients** box.

9. Complete the entries on the **Attributes** tab as explained in "Description of Attributes"
and click **Apply** to save the attribute entries.

10. Add other policy information:

- To add schedules, see "Adding New Schedules."

- To add templates or shell scripts, see "Specifying the List of Scripts."

- To add clients, see "Adding Clients to a Policy."

Description of Attributes

With a few exceptions, NetBackup manages a database backup like a file system backup. Policy attributes that are different for Oracle backups are explained below.

Your other policy attributes will vary according to your specific backup strategy and system configuration. Consult the *NetBackup System Administrator's Guide* for detailed explanations of the policy attributes.

Attribute	Description
Policy type	Determines the type of clients that can be in the policy and in some cases the types of backups that can be performed on those clients. To use NetBackup for Oracle on UNIX, you must have defined at least one Oracle policy.
Offhost backup method	To set up offhost backup, choose a method from the **Offhost backup method** pull-down menu. If you are configuring a backup (that does not require offhost backups), the method should be set to **Local Host**. If you use any of the offhost backup methods, follow the specific recommendations in the *NetBackup ServerFree Agent System Administrator's Guide* and *NetBackup for EMC Fastrax System Administrator's Guide*.
Policy storage unit	Storage unit to use for backups (for example, dlt or 8-mm tape). A storage device can attach to the master server or to a media server. Different policies can be configured to use different storage units. For example, one policy can be configured to use an 8-mm tape stacker (TS8) and another policy can be configured to use a DLT Library (DLT). If you use any of the offhost backup methods, follow the specific recommendations in the *NetBackup ServerFree Agent System Administrator's Guide* and *NetBackup for EMC Fastrax System Administrator's Guide*.
Keyword phrase	For NetBackup for Oracle on UNIX, the keyword phrase entry is ignored.
Allow frozen image clients	Select the checkbox to allow for NetBackup for Oracle ServerFree Agent's proxy backup operations. **Note** You must use the Clients tab to configure a frozen image method. See "Configuring a Frozen Image Method" in the *NetBackup ServerFree Agent System Administrator's Guide* and "Configuring a Policy for Fastrax" in the *NetBackup for EMC Fastrax System Administrator's Guide*.

Adding New Schedules

Each policy has its own set of schedules. These schedules control initiation of automatic backups and also specify when user operations can be initiated.

A Oracle backup requires at least two specific schedule types: an Application Backup schedule and an Automatic Backup schedule. You can also create additional schedules.

The following procedures explain how to configure the required schedule types, and how to add other new schedules.

▼ **To configure an Application Backup schedule**

1. Under the policy name, select **Schedules**.

2. In the right pane, double-click on the schedule named **Default-Application-Backup**.

A dialog box appears. The title bar shows the name of the policy to which you are adding the schedule.

All Oracle non-proxy database operations are performed through NetBackup for Oracle on UNIX using an Application Backup schedule. This includes those backups started automatically.

You must configure an Application Backup schedule for each Oracle policy you create. If you do not do this, you will not be able to perform a non-proxy backup. To help satisfy this requirement, an Application Backup schedule named Default-Application-Backup is automatically created when you configure a new Oracle policy.

3. Specify the other properties for the schedule as explained in "Schedule Properties."

The backup window for an Application Backup schedule must encompass the time period during which all NetBackup jobs, scheduled and unscheduled, will occur. This is necessary because the Application Backup schedule starts processes that are required for all non-proxy NetBackup for Oracle on UNIX backups, including those started automatically.

For example, assume that you:

- expect users to perform NetBackup operations during business hours, 0800 to 1300.

- configured automatic backups to start between 1800 and 2200.

The Application Backup schedule must have a start time of 0800 and a duration of 14 hours.

> **Tip** Set the time period for the Application Backup schedule for 24 hours per day, seven days per week. This will ensure that your NetBackup for Oracle on UNIX operations are never locked out due to the Application Backup schedule.

Example Settings for an Application Backup schedule.

Type of Backup	Schedule settings	Description	Settings
Application Backup	Retention	The length of time backup images are stored.	2 weeks
	Backup Window	The time during which a NetBackup operation can be performed.	Sunday through Saturday 00:00:01 - 23:59:59

▼ **To configure an automatic backup schedule**

 1. In the All Policies pane, expand the policy you wish to configure. Right-click **Schedules** and choose **New**.

 A dialog box appears. The title bar shows the name of the policy to which you are adding the schedules.
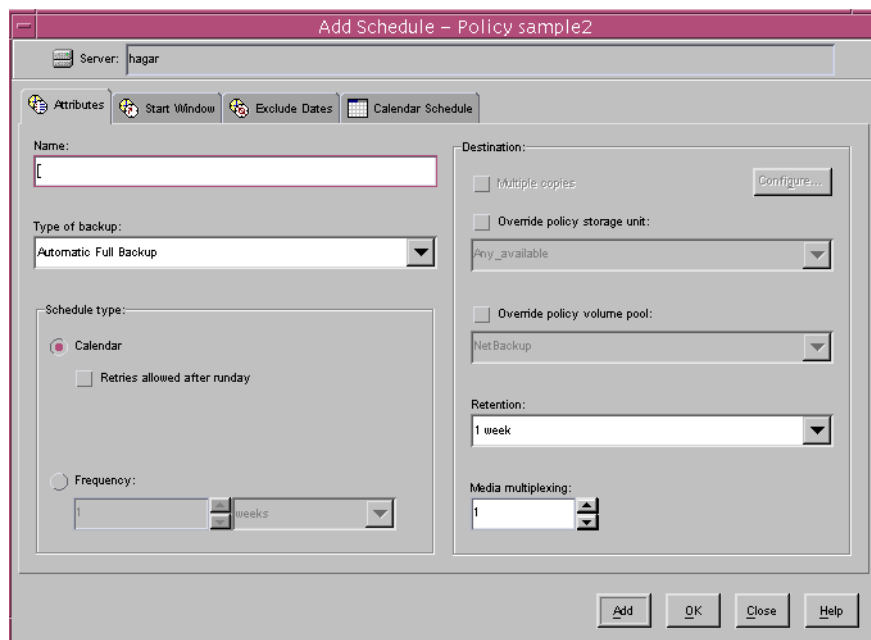
2. Specify a unique name for the schedule.

3. Select the **Type of Backup**.

For information on the types of backups available for this policy, see "Types of Backups."

4. Specify the other properties for the schedule as explained in "Schedule Properties."

When a schedule is executed, NetBackup sets environment variables based on which type of automatic backup schedule is selected. These environment variables can then be used to conditionally start a backup. Refer to "Environment Variables Set Up by NetBackup for Oracle on UNIX" on page 62.

To add other schedules, repeat step 1 through step 4.

Types of Backups

Application Backup	The Application Backup schedule enables user-controlled NetBackup operations performed on the client. At least one Application Backup schedule type must be configured for each Oracle policy. The Default-Application-Backup schedule is automatically configured as an Application Backup schedule.
Automatic Full Backup, Automatic Differential Incremental Backup, Automatic Cumulative Incremental Backup	An Automatic Full Backup, Automatic Differential Incremental Backup, or Automatic Cumulative Incremental Backup type schedule specifies the dates and times when NetBackup will automatically start backups by running the NetBackup for Oracle RMAN scripts in the order that they appear in the files/scripts list. If there is more than one client in the Oracle policy, the NetBackup for oracle scripts are executed on each client. **Note** For explanations of these backup terms, see "Oracle Terms" on page 5. This schedule is also responsible for running the actual proxy copy backups.

Schedule Properties

Some of the schedule properties have a different meaning for database backups than for a regular file system backup. These properties are explained below.

Other schedule properties will vary according to your specific backup strategy and system configuration. Consult the *NetBackup System Administrator's Guide* for detailed explanations of the schedule properties.

Property	Description
Type of backup	Specifies the type of backup that this schedule will control. The selection list shows only the backup types that apply to the policy you are configuring.
	For more information see "Types of Backups."
Frequency	This setting is used only for scheduled backups, and not for user-directed backups. **Frequency** specifies the period of time that will elapse until the next backup operation can begin on this schedule. For example, if the frequency is seven days and a successful backup occurs on Wednesday, the next full backup will not occur until the following Wednesday. Normally, incremental backups will have a shorter frequency than full backups.
Calendar	This setting is used only for scheduled backups, and not for user-directed backups. The **Calendar** option allows you to schedule backup operations based on specific dates, recurring week days, or recurring days of the month.

Property	Description
Retention	**Frequency based scheduling**
	The retention period for an Application Backup schedule refers to the length of time that NetBackup keeps non-proxy backup images.
	For non-proxy backups, the retention period for an Automatic Full Backup, Automatic Differential Incremental Backup, or Automatic Cumulative Incremental Backup schedule controls how long NetBackup keeps records of when scheduled backups have occurred.
	The NetBackup scheduler compares the latest record to the frequency to determine whether a backup is due. This means that if you set the retention period to expire the record too early, the scheduled backup frequency will be unpredictable. However, if you set the retention period to be longer than necessary, the NetBackup catalog will accumulate unnecessary records. *Therefore, set a retention period that is <u>longer</u> than the frequency setting for the schedule.*
	For example, if the frequency setting is set to one week, set the retention period to be more than one week.
	For proxy backups, NetBackup for Oracle ServerFree Agent uses the Automatic Full Backup, Automatic Differential Incremental Backup, or Automatic Cumulative Incremental Backup retention period to specify how long to keep proxy backup images.
	Note Oracle is not notified when NetBackup expires a backup image. You must use Oracle RMAN repository maintenance commands to periodically delete expired backup sets from the Oracle RMAN repository.
	Calendar based scheduling
	The retention period for an Application Backup schedule refers to the length of time that NetBackup keeps non-proxy backup images.
	For non-proxy backups, the retention period for an Automatic Full Backup, Automatic Differential Incremental Backup, or Automatic Cumulative Incremental Backup schedule controls how long NetBackup keeps records of when scheduled backups have occurred. However, this setting is not significant for calendar based scheduling.
	For proxy backups, NetBackup ServerFree Agent uses the Automatic Full Backup, Automatic Differential Incremental Backup, or Automatic Cumulative Incremental Backup retention period to specify how long to keep proxy backup images.
Multiple copies	If you are licensed for the Inline Tape Copy feature and wish to specify multiple copies on non-proxy backups for your Oracle policy, configure **Multiple copies** on the Application Backup schedule.
	For proxy backups, configure **Multiple copies** on the automatic backup schedule.

Expiration of Backup Files

NetBackup and Oracle each maintain a repository for backup image information. Currently, automatic expiration of backup images from both repositories is not supported.

The following describes how to expire images and synchronize the NetBackup and the Oracle repositories.

Expiration of Backup Images from the NetBackup Catalog Using Retention Level

NetBackup automatically controls the expiration of the Oracle backup images from its repository using the retention setting in a schedule.

For non-proxy backups, use the Application Backup schedule retention setting to specify the length of time before NetBackup expires a backup image.

For proxy backups, use the corresponding automatic backup schedule retention setting to specify the length of time before NetBackup expires a backup image.

When you use the retention setting to expire backup images, you must perform periodic Oracle repository maintenance to remove references to expired backup files.

Oracle Recovery Manager Catalog Maintenance

 Use RMAN repository maintenance commands to remove references to backup files. You can use these commands to delete backup image information from both the Oracle RMAN repository and the NetBackup repository. For more information on the RMAN repository maintenance commands, refer to "Maintaining the RMAN Repository" on page 84.

When a request is issued to delete a backup file from the RMAN repository, RMAN sends the request to NetBackup for Oracle ServerFree Agent to delete the corresponding image from the NetBackup repository, regardless of the retention level.

Specifying the List of Scripts

The File list in a database policy has a different meaning than the File list has for other policies. Normally, in a Standard policy, you would list files and directories to be backed up. But since you are now configuring a database policy, you will list templates or scripts.

For a discussion of NetBackup for Oracle templates and shell scripts, see "Creating Templates and Shell Scripts" on page 68.

Add templates or scripts only if you are setting up a policy for automatic scheduling. All templates or scripts listed in the Files list will be executed for the Automatic Full Backup, Automatic Differential Incremental Backup, or Automatic Cumulative Incremental Backup schedules as specified under the **Schedules** tab.

All templates or scripts specified in the Files list are executed during manual or automatic backups. NetBackup will start backups by running the templates or scripts in the order that they appear in the Files list.

Environment Variables

NetBackup sets some environment variables when it initiates an automatic backup schedule. A NetBackup for Oracle RMAN script can use these environment variables to determine what type of backup to perform.

For example, if an automatic cumulative incremental type of backup is executed, the NetBackup for Oracle RMAN script would perform a cumulative incremental backup. Only one NetBackup for Oracle RMAN script is necessary to handle different schedules.

For more details, refer to "Environment Variables Set Up by NetBackup for Oracle on UNIX" on page 62.

▼ **To add templates or scripts to the Scripts List**

Caution Be sure to use the correct NetBackup for Oracle RMAN script name in the file list to prevent an error or possibly a wrong operation.

1. In the left pane, click **Policies**.

2. In the All Policies pane, expand the policy you want to add the templates or shell scripts.

3. Right-click on **Files** and choose **New**.

The Add File dialog box appears. The title bar shows the name of the policy to which you are adding the scripts.
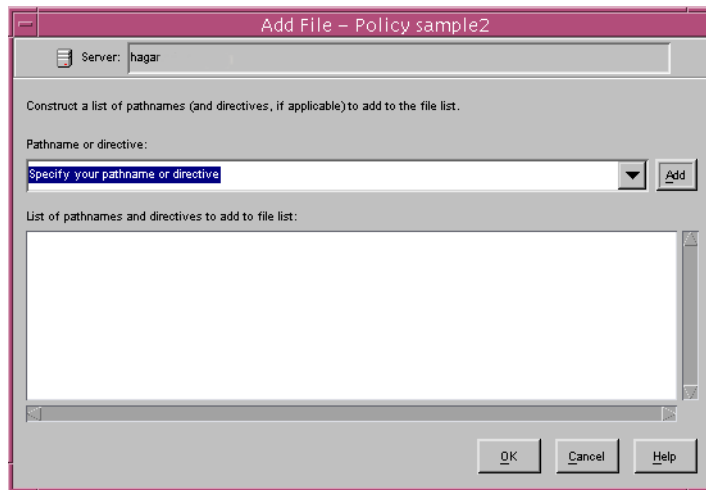
4. Type the name of the Oracle template or shell script.

Shell scripts:

Be sure to specify the full pathname when listing Oracle RMAN scripts. For example:

```
install_path/netbackup/ext/dbext/oracle/samples/rman/cold_database
_backup.sh
```

Be sure that the Oracle RMAN scripts listed here are installed on each client in the Client list.

Templates:

Since templates are stored in a known location on the master server, only the template filename should be entered. For example:

weekly_full_backup.tpl

5. Click **Add**.

6. To add more scripts, repeat step 4 and step 5.

Adding Clients to a Policy

The client list is the list of clients on which your NetBackup for Oracle on UNIX backups will be performed. A NetBackup client must be in at least one policy but can be in more than one.

▼ **To add clients to a policy**

1. In the left pane, expand **Policies**.

2. In the All Policies pane, expand the policy you wish to configure.

3. Right-click on **Clients** and choose **New**.

   The Add Client dialog box appears. The title bar shows the name of the policy where you are adding clients.
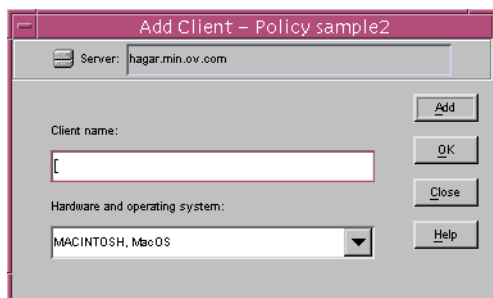
4. In the **Client name** text box, type the name of the client that you are adding.

   On the client the following should be installed:

   - Oracle
   - NetBackup client or server
   - NetBackup for Oracle on UNIX
   - the backup shell script(s) (unless you are using templates)

5. Choose the hardware and operating system type and click **Add**.

6. If this is the last client, click **OK**. If you are going to add more clients, repeat step 4 and step 5.

7. Configure Frozen Image backup methods for the client.

a. Select the client you want to configure for Frozen Image methods.

The Change Policy dialog box will change as follows.
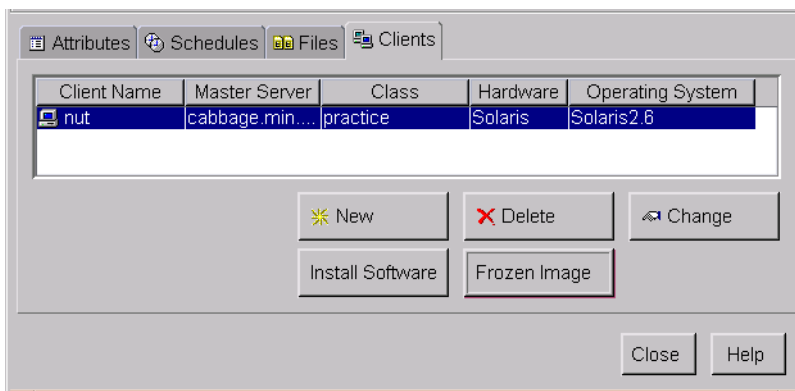
Note The **Frozen Image** button is enabled only if the **Allow frozen image clients** checkbox has been selected in the **Attributes** tab.

8. Click **Frozen image**.

The Frozen Image Client Configuration dialog appears.

Enter name of Frozen image source.

Select a frozen image method.

If nbu_snap is the frozen image method, enter the name of a raw partition to be used as cache.
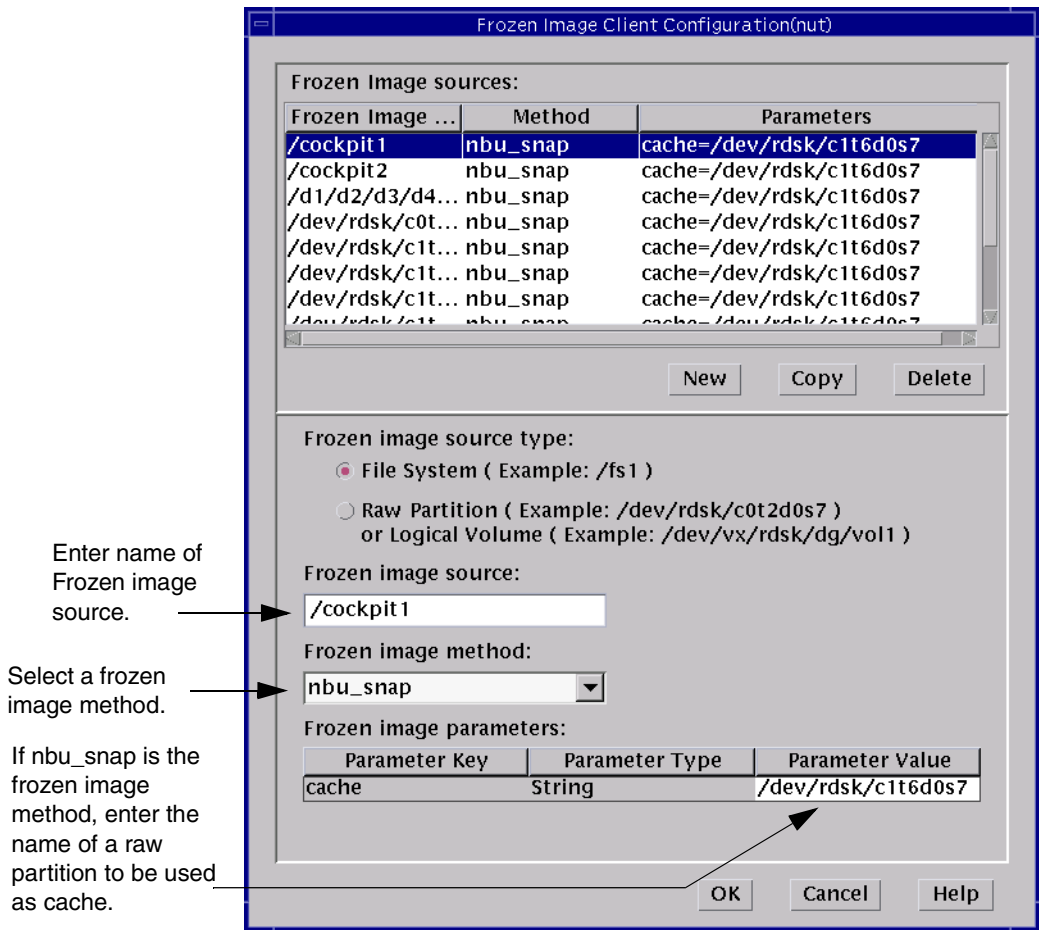
a. Select Frozen image source type (can be a File System, Raw Partition or Logical Volume).

b. Enter the name for the **Frozen image source** that contains the datafiles to be backed up.

c. Select a frozen image method.

Refer to the *NetBackup ServerFree Agent System Administrator's Guide* for more details on configuring a frozen image method for the client.

Configuring Run-Time Environment

The following is the order of precedence for the run-time configuration variable settings.

1. vendor-specific string with RMAN `send` command

Refer to "RMAN `send` Command and `parms` Operand" on page 65 for details.

2. RMAN `parms` operand

Refer to "RMAN `send` Command and `parms` Operand" on page 65 for details.

3. environment variables

Refer to "Environment Variables Set Up by NetBackup for Oracle on UNIX" on page 62 and to "Environment Variables Set Up by a User" on page 63 for details.

4. user `bp.conf`

Refer to "Creating `bp.conf`" on page 66 for details.

5. master `bp.conf`

Refer to "Creating `bp.conf`" on page 66 for details.

Environment Variables Set Up by NetBackup for Oracle on UNIX

When a schedule executes, NetBackup sets environment variables for templates and shell scripts to use when performing a NetBackup operation. These variables can be used to perform conditional operations. These variables are set only if the backup is started from the server (either automatically by the NetBackup scheduler or manually through the administrator interface).

NB_ORA_PC_SCHED

Name of the Oracle schedule used for a proxy copy backup (Automatic Full backup type).

NB_ORA_PC_BTYPE

NetBackup sets this variable to FI if the frozen image attribute of the Oracle policy is selected.

Note If this variable is not set to FI, NetBackup for Oracle ServerFree Agent will not allow you to initiate a proxy backup. RMAN will try to start regular backups if the proxy operand (versus the proxy only operand) of the backup command is specified.

NB_ORA_SERV

Name of the NetBackup Server.

NB_ORA_POLICY

Name of the Oracle policy.

NB_ORA_CLIENT

Name of the NetBackup client that hosts the Oracle database.

NB_ORA_FULL

Set to 1 if this is an Automatic Full Backup.

NB_ORA_INCR

Set to 1 if this is an Automatic Differential Incremental Backup.

NB_ORA_CINC

Set to 1 if this is an Automatic Cumulative Incremental Backup.

Refer to the following examples:

◆ "Example 1, cold_database_backup.sh" on page 119

◆ "Example 2, hot_database_backup.sh" on page 126

◆ "Example 3, cold_duplex_database_backup_full.sh" on page 132

◆ "Example 4, hot_database_backup_proxy.sh" on page 140

Environment Variables Set Up by a User

Environment variables set by the user will take precedence over those set up by NetBackup for Oracle.

The NetBackup for Oracle on UNIX variables that follow can be set for use in the Oracle user environment.

NB_ORA_SERV

Specifies the name of NetBackup master server.

NB_ORA_CLIENT

Specifies the name of the Oracle client. It is especially useful for redirecting a restore to a different client.

NB_ORA_POLICY

Specifies the name of the policy to use for the Oracle backup.

NB_ORA_SCHED

Specifies the name of the *Application Backup* schedule to use for the Oracle backup.

NB_ORA_PC_SCHED

Specifies the name of the schedule to use for the Oracle proxy backup (Automatic Full backup type). To insure that the backup uses the correct schedule for proxy backups, use the `send` command as explained in "RMAN `send` Command and `parms` Operand" on page 65.

NB_ORA_PC_BTYPE

Set this variable to FI to initiate proxy backups using the frozen image method. To insure that the backup uses the correct variable, use the `send` command as explained in "RMAN `send` Command and `parms` Operand" on page 65.

NB_ORA_PC_FILESYSTEMS

Specifies the number of file systems (raw partitions or logical volumes) that NetBackup for Oracle ServerFree Agent will back up concurrently in each proxy copy session.

The default value is equal to the number of target file systems. Target file systems contain the datafiles to be backed up.

Use the RMAN `send` command to pass this variable as explained "RMAN `send` Command and `parms` Operand" on page 65.

NB_ORA_PC_STREAMS

Specifies the number of backup streams per file system that NetBackup for Oracle ServerFree Agent starts simultaneously in each proxy copy session.

When a backup starts, NetBackup for Oracle ServerFree Agent groups all datafiles by the file system. Based on the value of this variable, it then organizes the files into a specified number of backup streams based on the file size. NetBackup for Oracle ServerFree Agent attempts to create streams of equal size.

The default value for NB_ORA_PC_STREAMS is 1.

Use the RMAN send command to pass this variable as explained "RMAN send Command and parms Operand" on page 65.

Note A backup using the Third-Party Copy Device offhost backup method is incompatible with multiplexing (refer to the *NetBackup ServerFree Agent System Administrator's Guide*). To avoid having multiple backup streams, set NB_ORA_PC_FILESYSTEM=1 and NB_ORA_PC_STREAMS=1.

If you want to configure multiple backup streams with the Third-Party Copy Device method, make sure that you have enough tape drives available for each stream.

Multiple Stream Configuration Example

In this example there are three file systems that contain Oracle datafiles:

FS1, FS2, FS3

A backup is configured using the following values:

NB_ORA_PC_FILESYSTEMS =2

NB_ORA_PC_STREAMS =3

These settings allow three backup streams per file system. Only two file systems can be backed up at a time. As a result, the following sequence will take place.

1. NetBackup for Oracle ServerFree Agent will initiate parallel backups of FS1 and FS2. Each file system will have three backup streams. Overall, there will be six backup jobs running at the same time.

2. When the backups of the datafiles on FS1 and FS2 are finished, the agent will initiate the backup of the datafiles on FS3, which will have three backup streams running at the same time.

RMAN `send` Command and `parms` Operand

Note	When connecting to a database using a TNS alias, you must use a send command or a parms operand to specify environment variables. In other words, when accessing a database through a listener, the environment variables set at the system level are not visible when RMAN is running.

Templates:

With templates, NetBackup for Oracle on UNIX environment variables are specified on the NetBackup for Oracle Configuration Variables Wizard page. See "Creating RMAN Templates Using the NetBackup for Oracle Backup Wizard" on page 69.

Shell scripts:

Use the send command or the parms operand to specify Netbackup for Oracle on UNIX environment variables for use during a backup or restore.

The send command sends a vendor-specific quoted string to NetBackup for Oracle on UNIX.

For example, the following will specify what policy and server to use for a database backup.

```
run {
   allocate channel t1 type 'sbt_tape';
   allocate channel t2 type 'sbt_tape';
    send 'NB_ORA_POLICY=your_policy, NB_ORA_SERV=your_server' ;
    send 'NB_ORA_PC_FILESYSTEMS=number_of_filesystems' ;
    send 'NB_ORA_PC_STREAMS=number_of_streams' ;
    send 'NB_ORA_PC_BTYPE=FI';
   backup proxy
   (database format 'bk_%U_%t');
}
```

Specify the variables in the string in the RMAN script after all channels have been allocated and before the backup command.

◆ Specify variables in the string on the RMAN command line to send them to all allocated channels.

Example 1:

See Example 4, hot_database_backup_proxy.sh on how to use it in your Oracle script.

Example 2:

To call RMAN from the UNIX prompt:

${ORACLE_HOME}`/bin/rman nocatalog send "'NB_ORA_POLICY=`*your_policy*`,`
`NB_ORA_PC_SCHED=`*your_proxy_copy_schedule*`,NB_ORA_PC_STREAMS=`*number_of_streams*`,`
`NB_ORA_PC_FILESYSTEMS=`*number_of_file_systems*`, NB_ORA_PC_BTYPE=FI'"`

The `parms` operand can also be used to set environment variables at runtime.

The following example uses the `parms` operand to specify what policy and server to use for a database backup. `parms` is set with each allocate channel command in the shell script.

```
run {
    allocate channel t1 type 'sbt_tape'
    parms="ENV=(NB_ORA_POLICY=your_policy, NB_ORA_SERV=your_server)";
    allocate channel t2 type 'sbt_tape'
    parms="ENV=(NB_ORA_POLICY=your_policy, NB_ORA_SERV=your_server)";
    backup
    (database format 'bk_%s_%p_%t');
}
```

For more information on the `send` command and `parms` operand of an `rman` command, see the *Oracle Recovery Manager User's Guide and Reference*.

Creating `bp.conf`

A NetBackup for Oracle on UNIX user can create an Oracle client `bp.conf` file in the Oracle user's home directory on the NetBackup for Oracle on UNIX client host. When a NetBackup for Oracle on UNIX operation is started, the user's `bp.conf` file is searched before the master configuration file (*install_path*`/netbackup/bp.conf`) on the client. Any option found at the user level takes precedence over the same option in the master level configuration.

The following options can be set in the user's `bp.conf` file:

BPBACKUP_POLICY
> This option specifies the name of the policy to use for the Oracle backup.

BPBACKUP_SCHED
> This option specifies the name of the Application Backup type of schedule to use for the Oracle backup.

CLIENT_NAME
> This option specifies the name of the Oracle client. It is especially useful for a redirected restore operation.

CLIENT_READ_TIMEOUT
> This option specifies the number of seconds the Oracle client will initially wait for a response from the NetBackup server. The default value is 900.

SERVER

>This option specifies the name of the NetBackup master server.

VERBOSE

>This option causes NetBackup to include more information in its logs.

See the *NetBackup System Administrator's Guide - UNIX* for more details and default values.

The following shows example `bp.conf` entries for an Oracle user:

```
SERVER=jupiter
CLIENT_READ_TIMEOUT=1200
VERBOSE=1
```

Creating Templates and Shell Scripts

RMAN templates and RMAN shell scripts contain commands that are used to execute NetBackup for Oracle ServerFree Agent backup and recovery jobs. Templates and scripts must be created before NetBackup for Oracle ServerFree Agent can perform scheduled backups. These are the template files or shell scripts that are specified in policy configuration on the NetBackup server.

For more information on scripts, see the *Oracle Recovery Manager User's Guide and Reference*.

Recovery Manager (RMAN)

RMAN Templates and Shell Scripts

Templates:

The NetBackup for Oracle Backup Wizard is used to create backup templates. This wizard is initiated from the NetBackup Backup, Archive, and Restore interface. See "Creating RMAN Templates Using the NetBackup for Oracle Backup Wizard" on page 69.

The NetBackup for Oracle Backup Wizard does not support all of the RMAN commands and options provided by Oracle. A shell script should be written for situations where a template does not provide all of the required functionality.

Shell scripts:

Shell scripts are written by the user and must conform to RMAN and UNIX shell syntax. Sample backup and recovery shell scripts are installed on the client with the NetBackup for Oracle agent. Modify these scripts to meet your individual requirements. See "Creating RMAN Scripts Manually" on page 78 for more information on sample scripts.

NetBackup for Oracle also provides a utility, bpdbsbora, that can generate a shell script from a Backup Wizard template. This allows a user to create a template with the Wizard, then generate a shell script from it. The user can then run the shell script, or modify the shell script further. See "Creating RMAN Shell Scripts Using bpdbsbora" on page 78.

Storing RMAN Templates and Shell Scripts

Templates:

The NetBackup for Oracle Backup Wizard saves a backup template to a NetBackup specific location on the current NetBackup master server. A backup template is retrieved from the master server as part of a backup (server-directed, scheduled, or user-directed) and is executed on the client. Backup templates are associated with a policy by specifying its name in the policy file or script list. Because backup templates are stored on the server in a known location, server directed and scheduled backups will use the same copy of the template for each client in the policy client list.

The NetBackup for Oracle Recovery Wizard saves a restore template to a user specified location on the client. The location specified should include a fully qualified path to a directory where the user has write access. For information about the recovery wizard, see "Using the NetBackup for Oracle Recovery Wizard" on page 95.

Templates store encrypted passwords that are decrypted at runtime.

Shell scripts:

RMAN shell scripts must reside on the NetBackup client. Backup shell scripts are associated with a policy by specifying the file name (including path) in the policy file or script list. This means that for server-directed or scheduled backups, each client in the policy's client list must have a copy of the script with the same name in the same location. See "Specifying the List of Scripts" on page 55.

The backup and recovery processes sometimes require passwords for Oracle database access and/or system user accounts. RMAN shell scripts, because a Shell interprets them, store passwords in clear text.

Creating RMAN Templates Using the NetBackup for Oracle Backup Wizard

NetBackup for Oracle includes a Backup Wizard that solicits information about desired RMAN backup operations. The wizard uses the information to create a template that can be run immediately or saved in a NetBackup specific location on the current master server for later use.

Please review your Oracle Backup and Recovery Guide for more information on backup strategies and RMAN functionality.

❖   To start the NetBackup Backup, Archive, and Restore interface, execute:

> *install_path***/netbackup/bin/jbpSA &**

If NetBackup for Oracle is installed, the Backup, Archive, and Restore interface for your Oracle client displays an Oracle node in the left pane. Click the **Backup Files** tab, and expand the Oracle node in the left pane to view an Oracle instance hierarchy. Select a node in the left pane to view details in the right pane.

If your current login does not have Oracle SYSDBA privileges, the following dialog appears when you select an instance to expand:
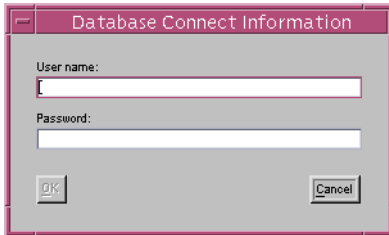
Enter your Oracle database login User name and Password with SYSDBA privileges to continue.

▼ **To use the Backup Wizard**

1. In the left pane of the Backup, Archive, and Restore interface, select the Oracle instance. In the right pane, select the Oracle objects to back up. Click **Backup**.
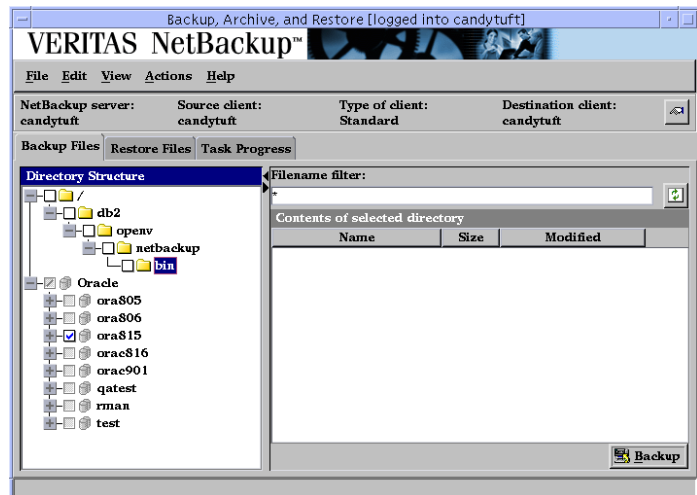
The Welcome
screen appears.
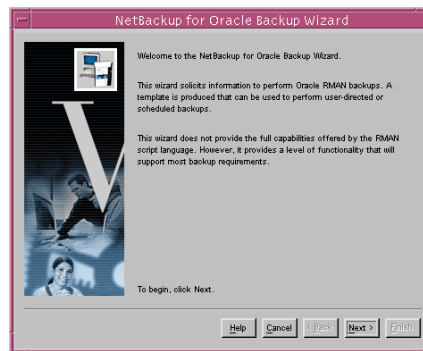
Click **Next** to continue.

2. Database administrators
perform backup
operations. Choose either
operating system
authentication or password
files to authenticate
database administrators.

The recovery catalog is a
repository of information
that is used and maintained
by RMAN. You are not
required to use a recovery
catalog, but Oracle
recommends it.
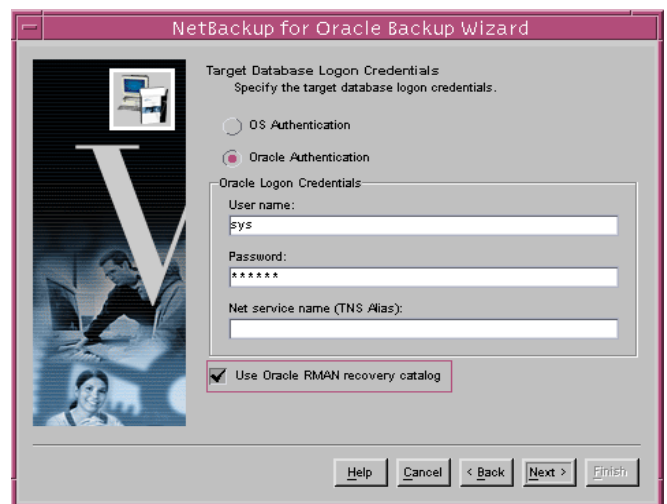
Click **Next** to continue.

Note Only one Oracle SID can be specified at any one time in the environment. Therefore,
if you are using a recovery catalog and thereby multiple Oracle SIDs, you will want
to use a Net Service Name for either the target database (in Step 2) or the recovery
catalog database (in Step 3). See the Oracle Net Administrator's Guide for more
information on Net Service Names.

3. The Recovery Catalog consists of a set of Oracle tables and views used by Recovery Manager to manage the backup, restore, and recover of Oracle databases. The recovery catalog schema should not be set up on the same Oracle SID as the target database. The **User name**, **Password**, and **Net service name (TNS alias)** together make up the database connect string. Click **Next** to continue.
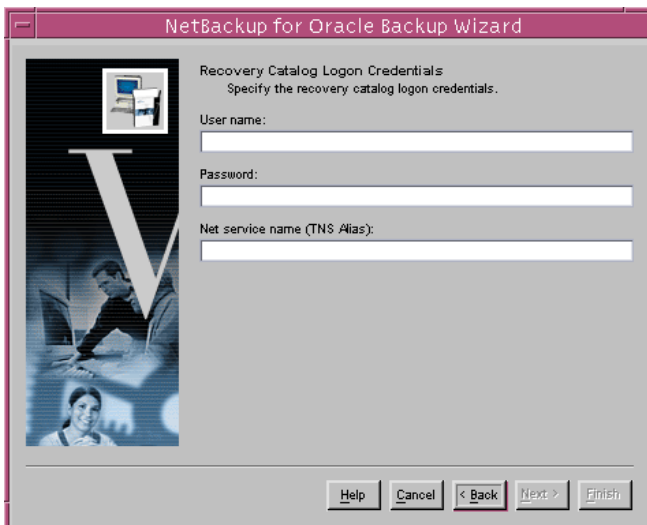
4. An archived redo log is a copy of an online redo log that has been copied to an offline destination. If the database is in ARCHIVELOG mode, Oracle copies each online redo log as it is filled. Select **Include Archived Redo Logs in Backup** to back up the archived redo logs.

Select **All** to back up all of the archived redo logs.

Select **Range** to specify which archived redo logs to include in the backup. You can also opt to **Delete archived redo logs after they are backed up**.

Click **Next** to continue.
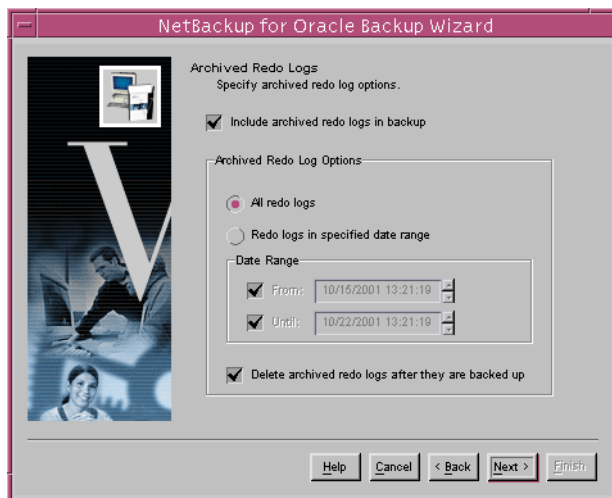
5. Configuration refers to a set of parameters used as default values in subsequent wizard pages. You have the choice of using a **Default Configuration** supplied by NetBackup or an **Existing template configuration** by supplying a **Template name**.

The **Use an existing configuration** option allows you to:

- use the same configuration at different times

- make minor changes to an existing configuration without having to start from scratch

- create a configuration similar to an existing configuration without having to define it from scratch.

Click **Next** to continue.
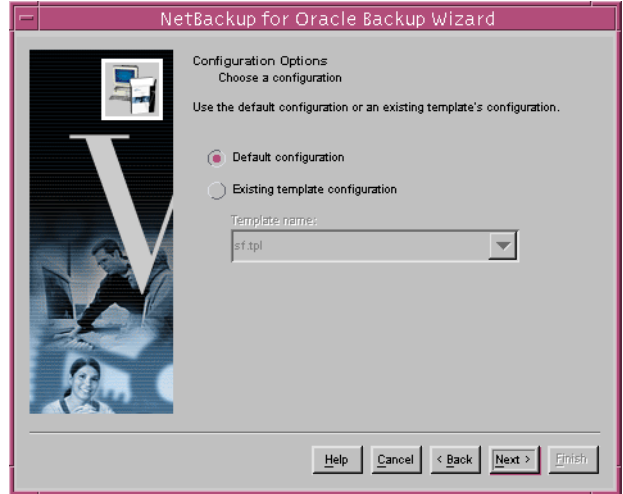
6. A backup set is a logical structure containing one or more physical backup pieces, such as datafiles or archived logs. Each backup piece must have a unique name.

Backup filename format specifies the file name format to use for the backup pieces created. **Backup set identifier** optionally specifies an identifier for the backup set(s). This is typically a meaningful name like "weekly_full_backup."
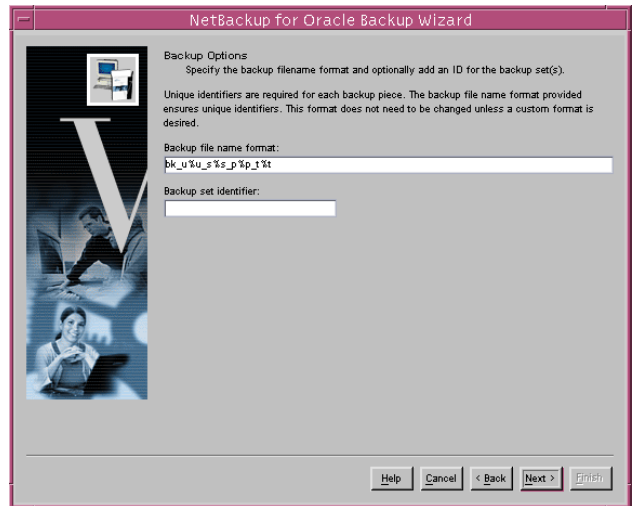
The following substitution variables are available to aid in the generation of unique backup piece names:

%p   Specifies the backup piece number within the backup set

%s   Specifies the backup set number

%d   Specifies the database name

%n   Specifies the padded database name

%t   Specifies the backup set time stamp

%u   Specifies an 8 character name composed of compressed representations of the backup set number and the time the backup set was created.

The following are valid only with Oracle8.1 and later:

%c   Specifies the copy number of the backup piece within a set of duplexed backup pieces

%U   Specifies a shorthand for %u_%p_%c that guarantees uniqueness in generated backup file names

NetBackup recommends that a %t be placed at the end of the backup filename format. Without this timestamp, there may be performance degradation as the NetBackup catalog grows.

Caution   Create a filename format that will generate unique filenames. In Oracle 8.0.x, non-unique filenames can cause backup data to be overwritten, resulting in possible loss of data. In later Oracle versions, non-unique filenames will result in error messages.

Click **Next** to continue.

7. Specify the database state. Choose the **OFFLINE BEFORE** option to shut down and start up the database in mount state before the backup. Choose the **ONLINE AFTER** option to shut down and start up the database in an open state after the backup completes successfully.

If you are starting the target database in the mount or open state, specify the **Oracle database initialization parameter file**.

Click **Next** to continue.
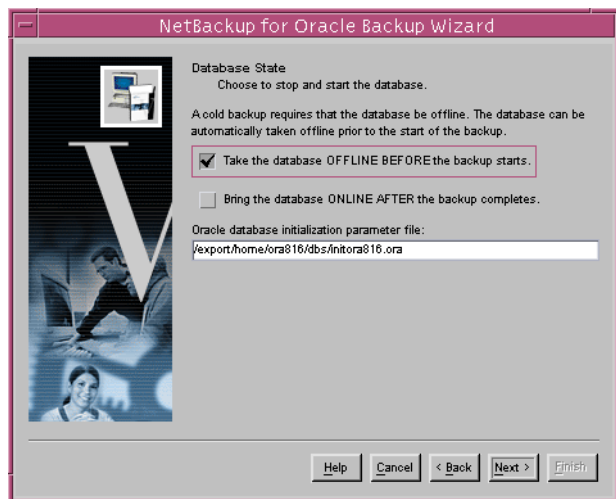
8. In a **Normal** Backup, RMAN controls data transfer between an Oracle database file and the storage device.

Choosing the **Proxy/Offhost (ServerFree)** method allows NetBackup for Oracle ServerFree Agent to control data transfer. The ServerFree Agent integrates the Oracle proxy copy feature with frozen image services and provides offhost backup of Oracle data files of Fibre Channel networks.
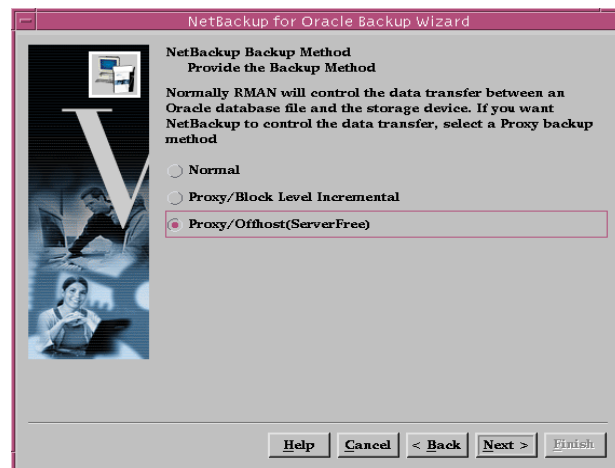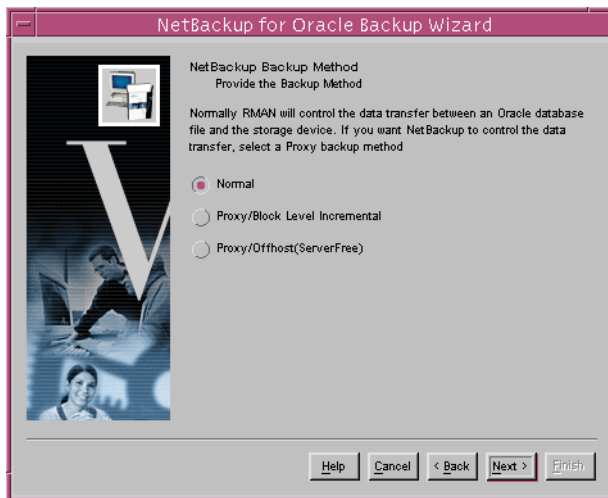
9. NetBackup normally executes a backup using configuration information defined in the policy and in configuration files. Use this screen to set or override certain configuration variables when the backup is started.

See "Configuring the Run-Time Environment" for more information on NetBackup configuration variables.
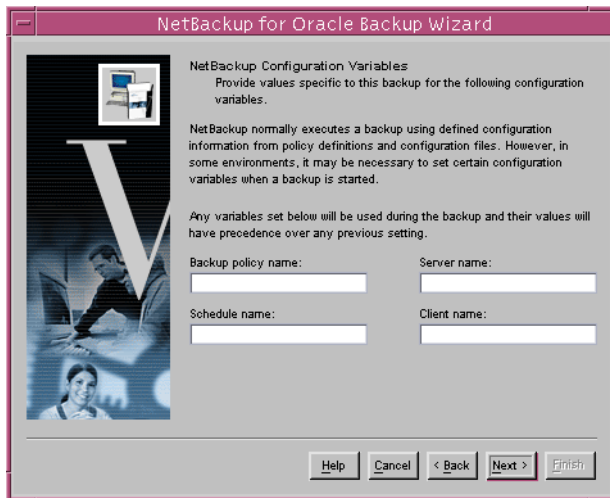
10. On the **Backup Limits** screen, set parameter values for non-proxy backup limiters.

Read rate specifies the maximum number of blocks per second that will be read from each input data file. **Size of a backup piece** sets the maxiumum size for each backup piece. **Number of open files** specifies the maximum number of input files that the backup will have open at any given time.
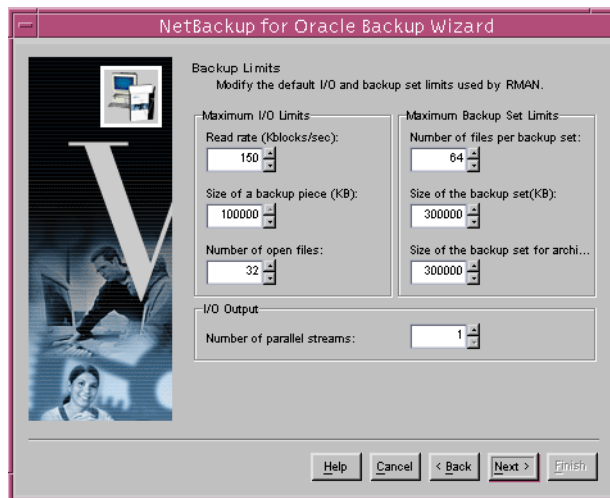
Number of files per backup set sets the maxiumum number of backup pieces to include in one backup set. Setting the **Size of the backup set** and the **Size of the backup set for archived redo logs** is useful if you want to make each backup set no larger than one tape.

The **Number of parallel streams** specifies the maximum number of connections between RMAN and a database instance. This value should be at least set to the number of storage devices used for the backup.

Click **Next** to continue.

11. The **Backup Limits (Proxy)** screen will appear only if a proxy backup method was selected. Set parameter values for backup limits used for a proxy backup.

Setting the **Size of the backup set for archived redo logs** is useful if you want to make each backup set no larger than one tape.

Modify **Override number of parallel file system backups** to change the number of file systems that the agent will

backup in parallel in each proxy copy session. The default value is equal to the number of target file systems.

Specify **Number of streams per file system** to change the number of backup streams per file system that the agent starts simultaneously in each proxy copy session.

Click **Next** to continue.
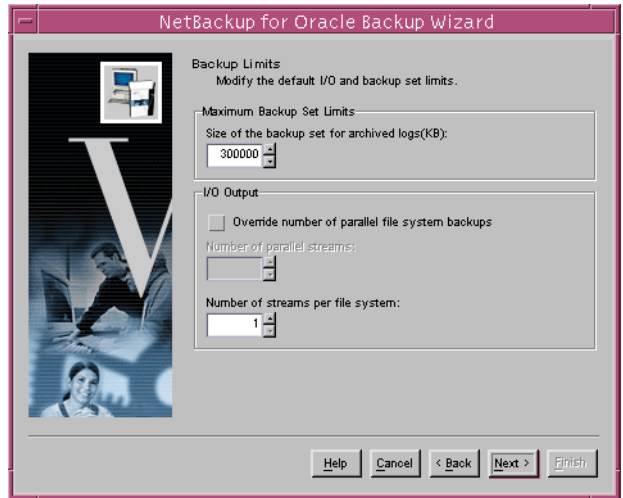
12. This screen displays a summary of the information you provided.

You can review the template in the **Template Summary** window, and use the **Back** button to find and change incorrect wizard entries.

If the template appears correct, you can run it immediately after finishing the wizard, or save it to a NetBackup specific location on the current master server.

Click **Finish** to exit the wizard.
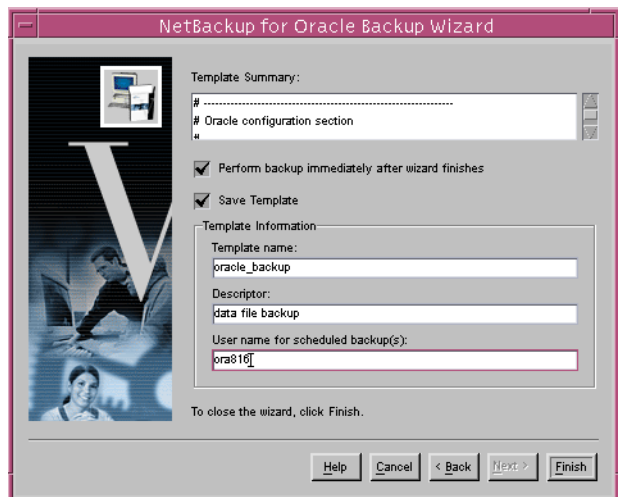
Creating RMAN Shell Scripts Using `bpdbsbora`

`bpdbsbora` is a NetBackup for Oracle command line utility used to initiate template based user-directed backup and recovery. This utility is also used to generate RMAN shell scripts from templates created by the Backup Wizard.

At the command prompt, type:

`bpdbsbora -backup -g <`*script file name*`> -t <`*template name*`>`

The following options are supported:

`-g` *script file name*	Generates a shell script from a template. Enclose *script file name* in quotes if it contains blanks. This option may not be used with the run option. See "Using `bpdbsbora`" on page 102.
`-t` *template name*	identifies the template.
	`bpdbsbora` will retrieve backup templates from a known location on the master server. Therefore, specify only the file name.

Creating RMAN Scripts Manually

When NetBackup for Oracle on UNIX was initially installed, example scripts were placed in the following directory:

install_path`/netbackup/ext/db_ext/oracle/samples/rman`

The Oracle example scripts installed are:

```
pit_database_restore.sh
hot_tablespace_backup.sh
hot_database_backup.sh
database_restore.sh
cold_duplex_database_backup_full.sh
cold_database_backup.sh
hot_database_backup_proxy.sh
hot_tablespace_backup_proxy.sh
```

The following scripts show how to perform proxy backups of a database:

◆ `hot_database_backup_proxy.sh`

◆ `hot_tablespace_backup_proxy.sh`

1. Copy the example scripts to a different directory on your client. Oracle scripts can be located anywhere on the client.

2. Modify each script for your environment.

3. Make sure the `su` command logs into the correct user.

If you do not include an `su` – *user* (*user* is Oracle administrator account) in your Oracle scripts, they will not run with the proper permissions and environment variables. The result will be problems with your database backups or restores.

Example NetBackup for Oracle scripts for Recovery Manager can be found in Appendix A of this document. See "NetBackup for Oracle RMAN Scripts" on page 119 .

Testing NetBackup for Oracle on UNIX Configuration Settings

After you have configured the master server for NetBackup for Oracle on UNIX, you should test the configuration settings. For a description of status codes, refer to the *NetBackup Troubleshooting Guide - UNIX*.

NetBackup Administration Console for UNIX

Use this procedure to test a policy configuration on the NetBackup Administration Console for UNIX.

▼ **To test the configuration settings on a UNIX server**

1. Log onto the server as root.

2. Start the NetBackup Administration Console.

3. In the left pane, click **Policies**.

   The right pane splits into an All Policies pane and a details pane.

4. In the All Policies pane, click the policy you wish to test.

5. From the **Actions** menu, click **Manual Backup**.

   The Manual Backup dialog box appears.
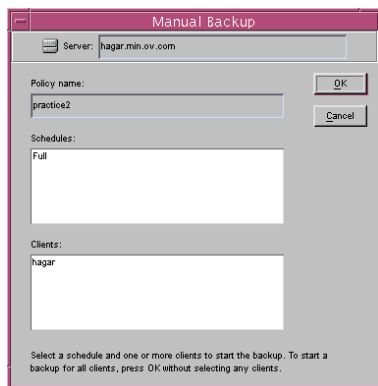
The Schedules pane contains the name of a schedule (or schedules) configured for the policy you are going to test. The Clients pane contains the name of the client(s) listed in the policy you are going to test.

6. Follow the instructions on the dialog box.

7. Click **Activity Monitor** on the NetBackup Administration Console.

When the Activity Monitor indicates job completion, check the output of the script(s) indicated in the policy you tested. The script will indicate where the ouput is stored. It is usually in the same directory as the original script, and is similarly named.
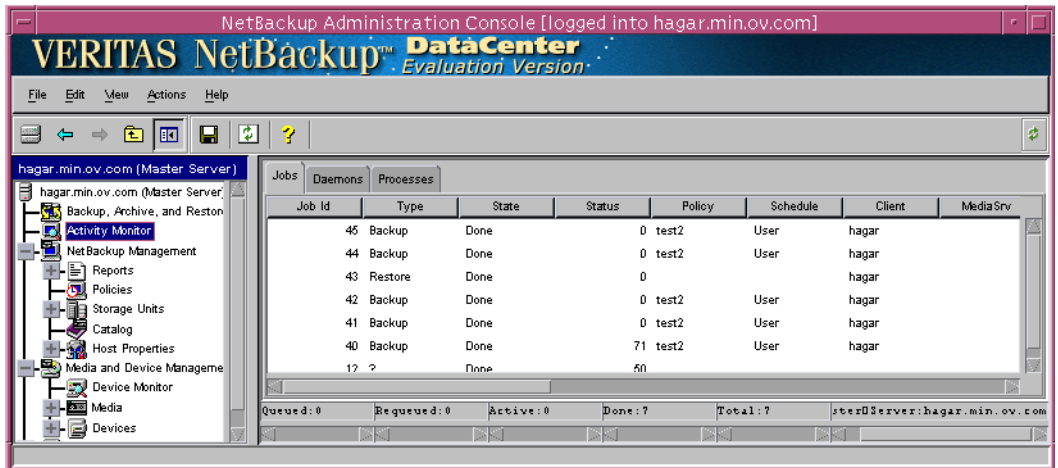
If the manual backup does not exit with a successful status (as indicated by the Activity Monitor and in the script output), refer to the Troubleshooting chapter.

Using NetBackup for Oracle ServerFree Agent 4

When installation and configuration is complete, you can use the NetBackup interfaces, the Oracle Enterprise Manager, or the command line interface to start Oracle backup and recovery, and to maintain the RMAN repository.

This chapter contains the following sections:

◆ Maintaining the RMAN Repository

◆ Querying the RMAN Repository

◆ Performing a Backup

◆ Browsing Backups

◆ Performing a Restore

◆ Performing Other RMAN Actions

Maintaining the RMAN Repository

The RMAN repository is the collection of metadata about your target databases that RMAN uses to conduct its backup, recovery, and maintenance operations. You can either create a recovery catalog in which to store this information or let RMAN store it exclusively in the target database control file. Although RMAN can conduct all major backup and recovery operations using just the control file, some RMAN commands function only when you use a recovery catalog. The following is a subset of repository maintenance commands provided by RMAN. Some of these commands may not be available with all versions of RMAN. Refer to your Oracle Backup and Recovery Guide for a complete description of maintenance commands for your version of RMAN.

Function	Description
Registering a database with the recovery catalog	Before using RMAN, you must register the target database in the recovery catalog. To do this, start and mount the target database but do not open it. At the RMAN prompt, issue a `register database` command.
Resetting the information in the recovery catalog	The `reset database` command directs RMAN to create a new database incarnation record in the recovery catalog.
Cross checking the information in the RMAN repository	Because the Media Manager can mark tapes as expired, the RMAN repository can contain outdated information. To ensure that data in the recovery catalog or control file is in sync with data in the media management catalog, perform a cross check. Use the cross check feature to:

◆ Determine whether a backup set is available or expired.

◆ Delete any expired backup sets.

◆ Call the media manager about the status of a backup piece and then mark it as available or expired.

Use either the `change ... crosscheck` or `crosscheck backup` command to check the specified files. Note that these commands do not delete images or repository records, you must use separate commands for these operations

Function	Description
	The `change ... crosscheck` command will query the media manager to determine if a backup piece is available. If a backup piece is unavailable, RMAN will mark the backup piece as expired. If it was expired but is now available, RMAN will mark the backup piece as available. The command syntax is:
	`change backuppiece {`*primary-key-list*|*filename-list*|*tag*`}` `crosscheck;`
	`change backupset {`*primary-key-list*`} crosscheck;`
	The `crosscheck backupset` command will operate on available and expired backup pieces. RMAN will update their status with the result (available or expired).
	To crosscheck a database, start RMAN and connect to the target database. Also connect to the recovery catalog if one is being used. At the `rman` command prompt, issue the following commands:
	`allocate channel for maintenance type 'sbt_tape';`
	`crosscheck backupset of database;`
Deleting expired backups	The `delete expired backup` command will operate only on expired backup pieces found in the recovery catalog. RMAN will remove them from the recovery catalog.
	To delete expired backupsets of a database from the recovery catalog, start RMAN and connect to the target and recovery catalog databases. At the RMAN command prompt, issue the following commands:
	`allocate channel for maintenance type 'sbt_tape';`
	`delete expired backupset of database;`
	The crosscheck and delete backupset commands allow you to restrict the list of objects operated on to the specified device type (i.e., disk or tape), object type (i.e., archived logs or database files), and date range.
Resynchronizing the recovery catalog	RMAN compares the recovery catalog to either the current control file of the target database or a backup control file and updates it with information that is missing or changed.
	If you are running in ARCHIVELOG mode, resynchronize the recovery catalog regularly since the recovery catalog is not updated automatically when a log switch occurs or when a redo log is archived.

Function	Description
	You must also resynchronize the recovery catalog after making any change to the physical structure of the target database. As with log archive operations, the recovery catalog is not automatically updated when a physical schema change is made.
	The RMAN `backup`, `copy`, `restore`, and `switch` commands update the recovery catalog automatically when the target database control file is available and the recovery catalog database is available when any of these commands are executed.
	If the recovery catalog is unavailable when you issue `backup` or `copy` commands, you should resynchronize it manually.
	To resynchronized the recovery catalog, start RMAN and issue the `resync catalog` command.
Changing the availability of a backup set or file copy	You may periodically need to notify RMAN that the status of a backup set, backup piece, datafile copy, or archived redo log has changed. The RMAN `change` command enables you to make a variety of useful record changes.
	The `change ... uncatalog` command removes references to a backup piece, datafile copy, or archivelog from the recovery catalog. This command works only with a recovery catalog.
	The `change ... delete` command removes references to a backup piece, datafile copy, or archivelog from the control file and recovery catalog. It physically deletes the file. This command works with or without a recovery catalog.
	The `change ... crosscheck` command removes references to a backup piece, datafile copy, or archivelog from the control and recovery catalog when that file no longer exists. This command works with or without a recovery catalog.
	The `change ... unavailable` command marks a backup piece, datafile copy, or archivelog as unavailable. This command works only with a recovery catalog.
Validating the restore of backups	A restore validation executes a restore test run without actually restoring the files. Test the restore of the entire database or individual tablespaces, datafiles, or control files.
	Use `restore ... validate` when you want RMAN to choose which backups should be tested.
	Use `validate backupset` when you want to specify which backup sets should be tested.

Querying the RMAN Repository

RMAN allows you to generate a number of reports relevant for backup and recovery using the `report` and `list` commands. The `list` command lists the contents of the recovery catalog or control file, while the `report` command performs a more detailed analysis.

Use the report and list commands to determine what you have backed up as well as what you need to back up. The information is available whether or not you use a recovery catalog.

You can use the `report` command to answer questions, such as:

◆ Which files need a backup?

◆ Which files have not had a backup in a while?

◆ Which files are not recoverable due to unrecoverable operations?

◆ Which backup files can be deleted?

◆ What was the physical schema of the database at some previous point in time?

The `list` command queries the recovery catalog and control file and produces a listing of its contents. The primary purpose of the list command is to determine which backups are available. You can list:

◆ Backup sets containing a backup of a specified list of datafiles.

◆ Backup sets containing a backup of any datafile that is a member of a specified list of tablespaces.

◆ All backup sets or copies of all datafiles in the database.

◆ Backup sets containing a backup of any archivelogs with a specified name and/or within a specified range.

◆ Incarnations of a specified database or of all databases known to the recovery catalog.

Refer to your *Oracle Recovery Manager User's Guide and Reference* for more details on querying the RMAN repository.

Performing a Backup

Backups can be performed by using:

◆ Automatic Backup of an Oracle Policy

◆ Manual Backup of an Oracle Policy

◆ User-Directed Backup From the Client

  - Using xbp to perform backups

  - Executing the NetBackup for Oracle template on the client

    - Using RMAN Template Administration

    - Using bpdbsbora

  - Executing the NetBackup for Oracle shell script on the client

  - Executing rman on the Client

Automatic Backup of an Oracle Policy

The most convenient way to back up your database is to set up schedules for automatic backups. When the NetBackup scheduler invokes a schedule for an automatic backup, the NetBackup for Oracle RMAN scripts run:

◆ In the same order as they appear in the file list

◆ On all clients in the client list

The NetBackup for Oracle RMAN scripts start the database backup by calling rman with the command file.

When the backup is started through NetBackup, NetBackup for Oracle ServerFree Agent leaves error checking for RMAN. The rman command generates an error if it considers a command invalid, but allows any of the commands it normally considers valid to proceed. This means that by specifying the wrong script file name you could start an unintended operation.

Manual Backup of an Oracle Policy

The administrator can use the NetBackup server software to manually execute an automatic backup schedule for the Oracle policy. See the *NetBackup System Administrator's Guide for UNIX* for detailed instructions.

Refer to "Testing NetBackup for Oracle on UNIX Configuration Settings" on page 80 for instructions on initiating a manual backup of an Oracle policy.

User Directed Backup From the Client

This section describes the following procedures for performing user-directed backups:

◆ Using xbp to Perform a Backup

◆ Executing NetBackup for Oracle Templates

◆ Executing the NetBackup for Oracle Shell Script

◆ Executing rman

Using **xbp** to Perform a Backup

Note xbp is not supported on AIX platforms.

The following describes how to use xbp to back up your database. Refer to the NetBackup User's Guide - UNIX for detailed instructions on using xbp to back up the database.

1. Log in as the Oracle administrator or as root.

If you are already logged in under a different user account, use the su – command to change to the Oracle administrator.

2. Execute xbp on the client to which you want to backup a database.

   ***install_path*/netbackup/bin/xbp**

3. In the Directory to Search Box, type the path name of the location of the NetBackup for Oracle RMAN scripts. For example:

   ***install_path*/netbackup/ext/db_ext/oracle/scripts/**

4. From the File menu, click **Browse File System for Oracle Backup Scripts**.

5. Select the backup script from the Files pane.

6. On the Backup menu, click **Backup Database Using Selected Scripts**. The xbp_confirm dialog box appears.

7. Click **OK**.

A NetBackup process called bphdb starts the NetBackup for Oracle RMAN script on the client.

8. View the status of the script execution.

    a. On the Backup menu, click **Report Progress Of Backup...**. The xbp_progress dialog box will appear.

    b. Select the log file for your backup.

    The Contents of Selected Log File pane displays only the status of the script execution. A status =0 message indicates that the script was successfully completed. Go to Step 9 for a detailed status report. For a status other than 0, refer to the Troubleshooting section of this manual. For additional information, check the output of the script.

9. View the log file for the NetBackup operation.

    a. Change directories to the `bphdb` log directory.

    cd *install_path*/netbackup/logs/bphdb

    b. Open the log file with the tail option.

    tail -f log.mmddyy

A `Backup completed SUCCESSFULLY` message indicates a successfully completed NetBackup operation.

Executing NetBackup for Oracle Templates

Using RMAN Template Administration

The RMAN Template Administration dialog is available in the NetBackup Backup, Archive, and Restore interface.

Use this dialog to run, edit, delete, rename, and view existing backup templates. These are the templates created by the NetBackup for Oracle Backup Wizard and stored in a pre-determined location on the master server. See "Using the NetBackup for Oracle Backup Wizard."

▼ **To use RMAN Template Administration**

1. In the Backup, Archive, and Restore interface, click **Actions->Administer Oracle Templates**.

The RMAN Template Administration window appears:
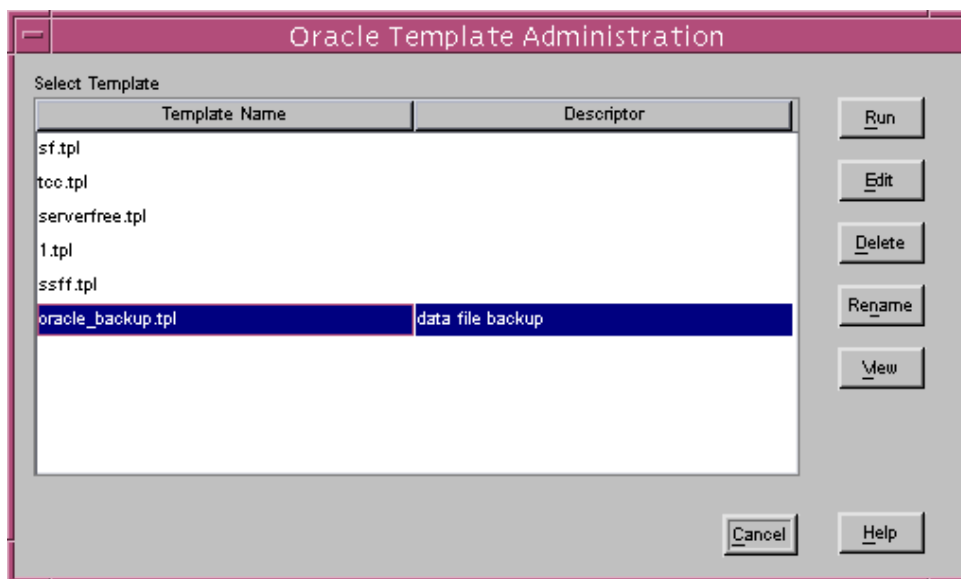
The Select Template list shows the names and descriptions of the RMAN backup templates stored on the current master server.

2. Select the name of the backup template you wish to run.

3. Click **Run**.

You can use the View Status tool to see the status of the backup. Click **Actions -> View Status**.

The RMAN Template Administration window provides the following functions:

Run	**Run** executes the selected template.
Edit	**Edit** is used to change the contents of an existing template. The selected template is loaded into the *NetBackup for Oracle Backup Wizard*.
Delete	**Delete** is user to delete the selected template.
Rename	**Rename** is used to change the name of the selected template. Only the creator of the template will be allowed to rename it.
View	**View** is used to see a summary of the selected template.

Using **bpdbsbora**

The bpdbsbora command allows you to run a backup template created by the NetBackup for Oracle Backup Wizard.

At the command prompt, type:

```
bpdbsbora -backup -r -t <template name>
```

where -r runs a template and -t identifies the template.

For example:

```
bpdbsbora -backup -r -t ORCL_Mon_full.tpl
```

bpdbsbora will retrieve backup templates from a pre-determined location on the master server. Therefore, specify only the file name.

Executing the NetBackup for Oracle Shell Script on the Client

If you know the pathname of the NetBackup for Oracle shell script that initiates the backup, you can execute the shell script from the UNIX command prompt.

For example, to perform a database backup, at the UNIX command prompt you might enter:

```
/oracle8/scripts/cold_database_backup.sh
```

The UNIX shell starts the database backup by executing the Oracle shell script. The Oracle shell script contains commands to execute .

The NetBackup for Oracle ServerFree Agent installation script installs sample scripts in the following location:

install_path/netbackup/ext/dbext/oracle/samples/rman/

Executing **rman**

As an Oracle user you can also execute the rman command at the UNIX command prompt with the RMAN command file as a parameter.

The following describes how to set the master server to windows and the Oracle Policy to obk before starting the backup.

rman

Since we are connecting to a database using a TNS alias, we must use the RMAN send command or a parms operand to specify the environment variables. At the command prompt enter:

```
rman target 'internal/oracle@ORCL' rcvcat 'rman/rman@RCAT'
send "'NB_ORA_POLICY=obk,NB_ORA_SERV=windows'" cmdfile
'/oracle/scripts/cold_database_backup_level0.rcv'
```

Note Use the NetBackup parameters NB_ORA_SERV, NB_ORA_CLIENT, NB_ORA_POLICY, and NB_ORA_SCHED to specify the NetBackup run-time configuration. Otherwise, the order of precedence for the run-time configuration variable settings is used. See "Configuring the Run-Time Environment" on page 63.

Browsing Backups

This section describes the following procedures for browsing backup image:

◆ Using xbp to Browse Backup Images

◆ Using the Recovery Catalog to Browse Backups

◆ Using bplist to Browse Backups

Using **xbp** to Browse Backup Images

Note xbp is not supported on AIX platforms.

The following describes how to use xbp to browse your database backups. Refer to the *NetBackup User's Guide - UNIX* for detailed instructions on using xbp to browse database backups.

1. Log in as the Oracle administrator or as root.

If a different user account is used, use the su command to change to the Oracle administrator.

2. Execute xbp on the client.

 ***install_path*/netbackup/bin/xbp**

The xbp dialog box appears.

3. From the File menu, click Configuration....

The xbp_config dialog box appears.

 a. In the Class type of image to browse, specify the Oracle policy.

 b. Click **OK** to close the dialog box.

    c. In the Directory To Search Box, enter / as the directory to search.

    d. Set the Range of Dates to include the date of your backup.

    e. Enter **9** in the Directory Depth Box.

    f. From the File menu, select Browse Backups (Restore). A list of backup images will appear in the dialog box.

Using the Recovery Catalog to Browse Backups

You can use the RMAN `report` and `list` commands to browse Oracle backups. Refer to "Maintaining the RMAN Repository" on page 84.

Using `bplist` to Browse Backups

You can use the `bplist` command to browse Oracle backups. The result is the list of backup file names. The following example uses `bplist` to search all Oracle backups for a client named jupiter.

```
install_path/netbackup/bin/bplist -C jupiter -t 4 -R /
/exb_n2bm5bco_1_1392342936
/exb_mabm02ko_1_1392170136
/exb_lqbltds6_1_1392083334
/block2/home/ora8/oradata/cabbage/drsys01.dbf
/block12/oradata/block12tbs2.dat
/block1/oradata/.block1tbsdf1new.dat
/block1/oradata/block1tbsdf1new.dat
```

The `-t 4` on this command specifies the Oracle backups. The `-R` specifies the default number of directory levels to search, 999. Refer to `bplist(1M)` man page for more information on this command.

The first three lines of output are non-proxy backup image names. The last four lines of output are proxy backup image names.

Performing a Restore

Make sure a backup has been successfully completed before attempting a restore. An error will occur if a backup history does not exist.

This section describes the following procedures for performing user-directed restores:

- ◆ Executing the NetBackup for Oracle template on the client
  - - Using the NetBackup Restore Wizard
  - - Using bpdbsbora
- ◆ Using xbp
- ◆ Executing the NetBackup for Oracle shell script on the client
- ◆ Executing rman on the client
- ◆ Redirecting a Restore to a Different Client
  - - Using rman

Executing the NetBackup for Oracle Template on the Client

Using the NetBackup for Oracle Recovery Wizard

NetBackup for Oracle includes a Recovery Wizard that solicits information from the user about the desired RMAN restore and recover operations. The wizard uses the information to create a template that can be used immediately or saved for later use.

The NetBackup for Oracle Recovery Wizard saves a recovery template locally, in a user-specified location on the NetBackup client. Recovery templates are not stored on the server because recovery is always user directed, not scheduled. Under normal circumstances, a recovery template will be run immediately and then deleted.

The recovery process sometimes requires passwords for Oracle database access and system user accounts. Templates store encrypted passwords that are decrypted at runtime.

Because recovery can be a complex process, it may be necessary to perform manual steps as part of the operation. Please review your Oracle Backup and Recovery Guide for more information.

Starting the NetBackup Backup, Archive, and Restore Interface

The NetBackup for Oracle Recovery Wizard is launched from the Backup, Archive, and Restore interface. You can access the interface through the NetBackup Administration Console, or from the command line.

❖ To start the interface from the NetBackup Administration Console, click **Backup, Archive, and Restore** in the left pane of the console.

❖ To start the NetBackup Backup, Archive, and Restore interface from the command line, execute:

*install_path***/netbackup/bin/jbpSA &**

The Restore Files tab

Click the **Restore Files** tab, and expand the Oracle node in the left pane to view an Oracle instance hierarchy. Select a node in the left pane to view details in the right pane.

Note that if the Oracle node is not visible, it's possible that your NetBackup for Oracle client does not have the appropriate policy type specified. Change the policy type with the following steps.

▼ **To change the client policy type**

1. On the **Actions** menu, select **Specify NetBackup Machines**.

2. On the Specify NetBackup Machines dialog, click the **Source client/Policy type** tab.

3. In the **Policy type** drop down list, select **Oracle**.

4. Click **OK**.

Using the Recovery Wizard

When you are ready to perform a recovery, follow these steps to create and run a template with the Recovery Wizard.

▼ **To use the Recovery Wizard**

1. In the Directory Structure pane of the Backup, Archive, and Restore interface, select the Oracle instance. In the right pane, select the database object you wish to restore. Click **Restore**.
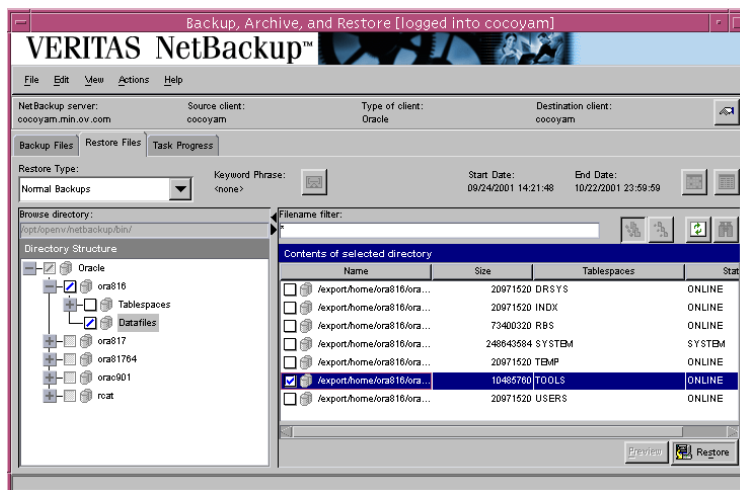
The Welcome screen appears.
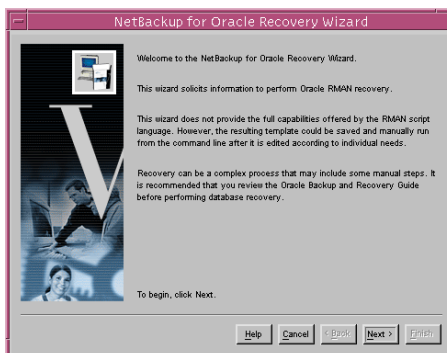
Click **Next** to continue.

2. Database administrators perform recovery operations. Choose either operating system authentication or password files to authenticate database administrators.

The recovery catalog is a repository of information that is used and maintained by RMAN. You are not required to use a recovery catalog, but Oracle recommends it.

Click **Next** to continue.
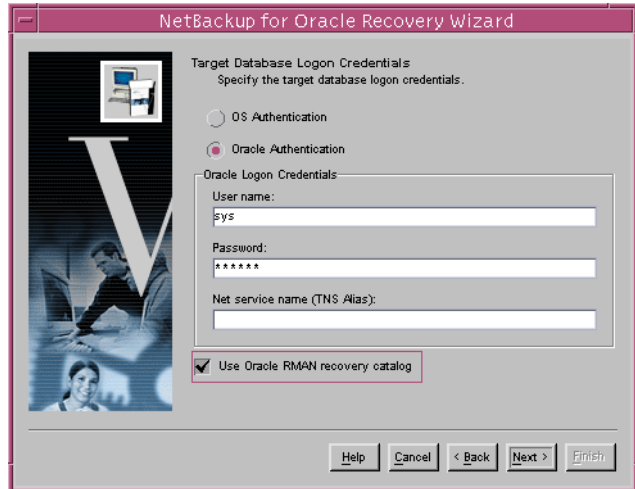
Note Only one Oracle SID can be specified at any one time in the environment. Therefore, if you are using a recovery catalog and thereby multiple Oracle SIDs, you will want to use a Net Service Name for either the target database (in Step 2) or the recovery catalog database (in Step 3). See the Oracle Net Administrator's Guide for more information.

3. The Recovery Catalog consists of a set of Oracle tables and views used by Recovery Manager to manage the backup, restore, and recover of Oracle databases. The recovery catalog schema must not be set up in the same Oracle SID as the target database. The **User name**, **Password**, and **Net service name (TNS alias)** together make up the database connect string.

Click **Next** to continue.
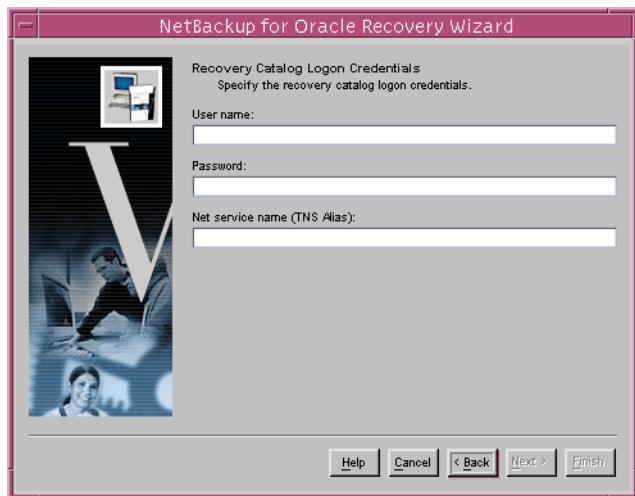
4. A restore operation retrieves and reconstructs physical database files from one or more backup sets. A recover brings the database files to a consistent state at a specific point in time, or to the last committed transaction. Normally you will want to both restore and recover.

RMAN by default does not restore read-only files. Checking the **Restore read-only files if they are not current** box will cause RMAN to restore any read-only files that do not meet all of the conditions for being current.

Specifying a backup set identifier overrides the RMAN default of restoring the most recent available backup.

RMAN by default does not recover read-only files. Checking the **Recover read-only files if they are not current** box will cause RMAN to recover any read-only files that do not meet all of the conditions for currency.

The option **Suppress the application of redo logs (apply incremental backups only)** is intended for recovery of NOARCHIVELOG databases using incremental backups. In this situation, if you do not specify this option when recovering a NOARCHIVELOG database, Oracle aborts and issues an error.

For both restore and recover, the **Number of parallel streams** specifies the maximum number of connections between RMAN and a database instance.

Click **Next** to continue.
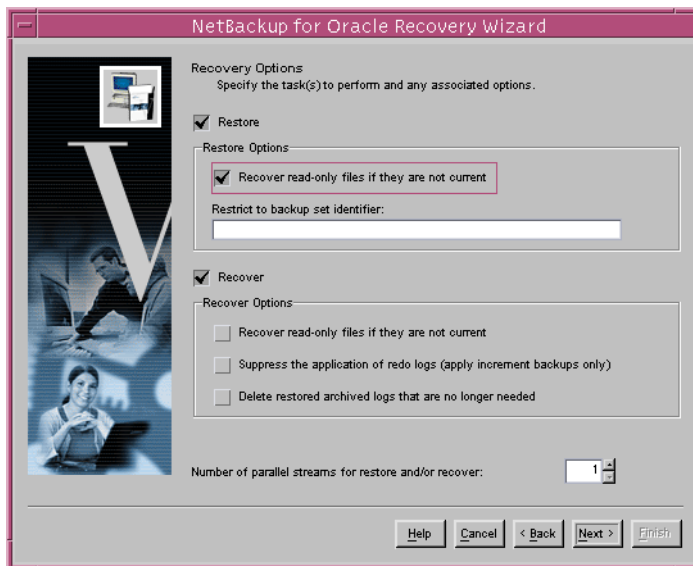
5. This screen appears if you chose **Restore** on the Recovery options screen.

When restoring for a point-in-time recovery, specify a backup set limiter.

If you use a redo log sequence number, indicate the thread number for the redo log in question.

Click **Next** to continue.
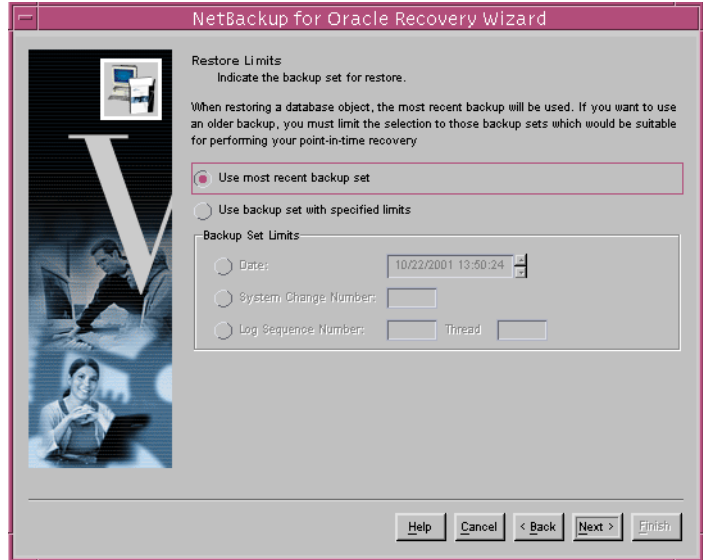
6. This screen appears if you chose **Recover** on the Recovery Options screen.

Choose to make the files current, or choose to recover the files to a specific point-in-time

Click **Next** to continue.
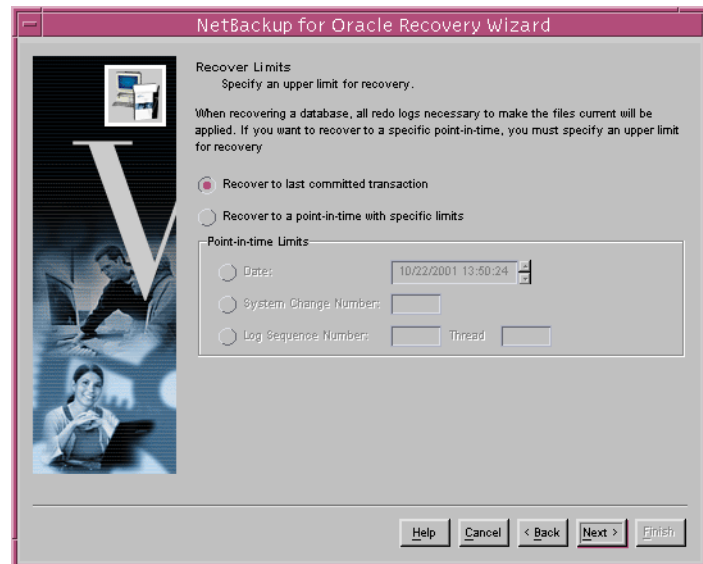
7. Specify the database state. Choose the **OFFLINE BEFORE** option to shutdown and start up the database in mount state before the restore. Choose the **ONLINE AFTER** option to shut down and start up the database in an open state after the restore completes successfully.

   If you are performing incomplete recovery, select **Reset logs on database open** to open the database with the RESETLOGS option.
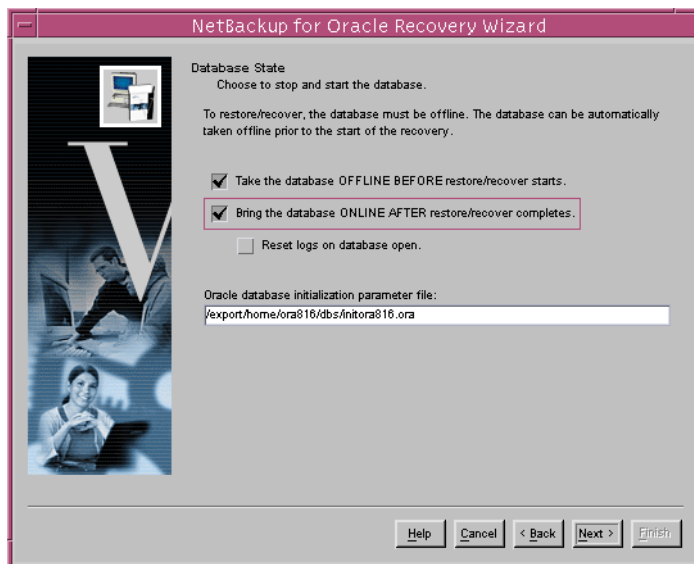
If you are starting the target database in the mount or open state, specify the **Oracle database initialization parameter file**.

Click **Next** to continue.

8. This screen displays a summary of the information you provided.

   You can review the template in the **Selection Summary** window, and use the Back button to find and change incorrect wizard entries.

   If the template appears correct, you can run it
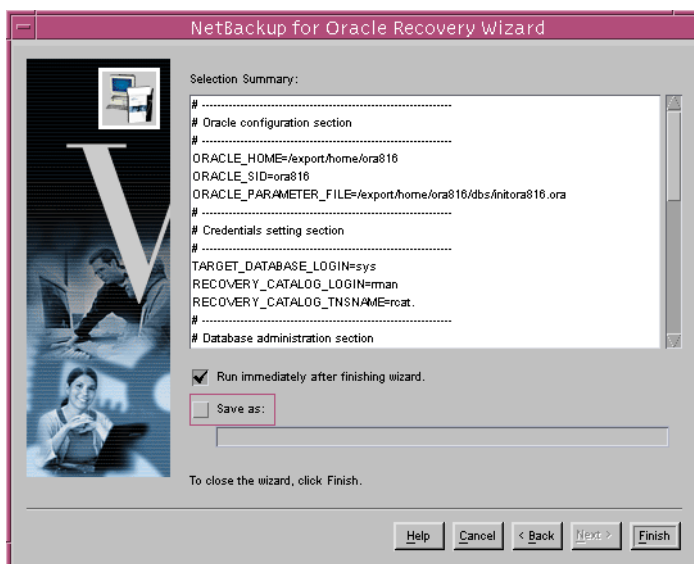
immediately after finishing the wizard, or save it to a user specified location on the client.

> **Note** The location specified should include a fully qualified path to a directory where the user has write access.

Click **Finish** to exit the wizard.

Using `bpdbsbora`

The `bpdbsbora` command allows you to run a recovery template created by the NetBackup Recovery Wizard.

At the command line, type:

/*install_path*/netbackup/bin/bpdbsbora -restore -r -t <template name>

where -r runs a template and -t identifies the template

For example:

bpdbsbora -restore -r -t /oracle/restore_templates/ORCL_MON_Full.tpl

Restore templates do not reside in a pre-determined location on the master server. They are considered to be temporary in nature and should reside on the client. If the full path is not specified as part of the restore template name, it must reside in the current directory.

Using xbp

The following describes how to use xbp to restore your database. Refer to the *NetBackup User's Guide - UNIX* for detailed instructions on using xbp to restore database backups.

1. Log in as the Oracle administrator or as root.

If a different user account is used, use the su– command to switch to the Oracle administrator.

2. Execute xbp on the client to which you want to restore a database.

install_path/netbackup/bin/xbp

3. In the Directory to Search Box, type in the path name of the location of the NetBackup for Oracle RMAN scripts. For example:

install_path/netbackup/ext/db_ext/oracle/scripts/

4. From the File menu, click Browse File System for Restore Scripts. The xbp dialog box appears.

5. Select the restore script from the Files pane.

6. On the Restore menu, click Restore Database Using Selected Scripts. The xbp_confirm dialog box appears.

7. Click **OK**.

A NetBackup process called `bphdb` starts the NetBackup for Oracle RMAN script on the client.

8. View the status of the script execution.

    a. On the Restore menu, click Report Progress Of Restore.... The xbp_progress dialog box will appear.

    b. Select the log file for your restore.

        The Contents of Selected Log File pane displays only the status of the script execution. A status =0 message indicates that the script was successfully completed. Go to Step 9 for a detailed status report. For a status other than 0, refer to the Troubleshooting section of this manual.

        Also, check the ouput from the restore script(s). The script will indicate where the output is stored. It is usually in thes ame directory as the original script and is similarly named.

9. View the log file for the NetBackup operation.

    a. Change directories to the `bphdb` log directory.

    cd /*install_path*/netbackup/logs/bphdb

    b. Open the log file with the tail option.

    tail -f log.mmddyy

A `Restore completed SUCCESSFULLY` message indicates a successfully completed NetBackup operation.

Executing an Oracle Script From a Command Line

If you know the pathname of the Oracle shell script that initiates the recovery, you can start it from the UNIX command prompt. For example, to perform a database recovery at the UNIX command prompt you might enter:

```
/oracle8/scripts/database_restore.sh
```

The UNIX shell starts the database restore by executing the Oracle shell script file. The Oracle shell script file contains commands to execute rman.

The NetBackup for Oracle ServerFree Agent installation script installs sample scripts in the following location:

install_path/netbackup/ext/dbext/oracle/samples/rman/

Executing `rman` on the Client

Execute the rman command from the UNIX command line on the client, by using the appropriate RMAN command file as a parameter. This example assumes that you are logged on as an Oracle administrator.

▼ **To execute** rman **on the client**

At the command prompt, enter:

```
rman target 'internal/oracle@ORCL' rcvcat 'rman/rman@RCAT'
  cmdfile '/oracle8/scripts/database_restore.rcv'
```

Redirecting a Restore to a Different Client

With NetBackup for Oracle ServerFree Agent you have the option to restore a database to a client other than the one that originally performed the backup. The process of restoring data to another client is called a redirected restore.

The following illustrates a typical redirected restore process.
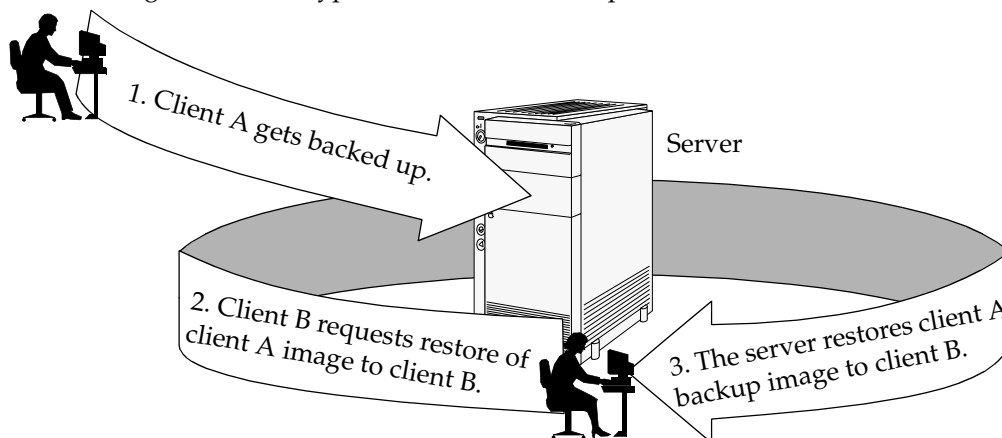
Note that the user on client A was not able to initiate a redirected restore to client B. Only the user on client B, the client receiving the backup image, could initate the redirected restore.

Note The same user name (UNIX account) that was used for the Oracle database backup must be used for the alternate client restore.

Server Configuration

Ensure that the NetBackup server is configured to allow redirected restore. The administrator can remove restrictions for all clients by creating the following file on the Netbackup master sever:

install_path/netbackup/db/altnames/No.Restrictions

Or the administrator can restrict clients to restore only from certain other clients by creating and then adding client names to the file:

install_path/netbackup/db/altnames/*client_name*

Where *client_name* is the name of the client allowed to do the redirected restore.

See the *NetBackup System Administrator's Guide* for details.

Using RMAN to Perform a Redirected Restore

Note The same user name (UNIX account) that was used for the Oracle database backup must be used for the alternate client restore.

Perform the following on the different client host if you want to restore RMAN backups that are owned by another client.

1. Enable a network connection to the RMAN catalog database, which was used by the original client.

Note If the RMAN catalog database has been lost, restore the catalog database first before continuing with the redirected restore.

2. Set the environment variable, NB_ORA_CLIENT, to the original client.

3. Make the `init.ora` file of the source client available to the destination client. You can do this by copying the file to the destination client, or by modifying the file on the destination client. Change all location-specific parameters.

4. Grant write permission to the directory where the datafiles will be restored.

5. Set up a password file for the destination client database.

6. Start up the database in the nomount state.

7. Start `rman`, connecting to the catalog.

8. Set *dbid* to be the dbid of the source client database.

9. Connect to the target database without using a userid/password.

10. Run an RMAN restore script or type the RMAN commands for the restore.

Example

In this example assume:

- Original client is camel
- Different client is giraffe
- Server is lion
- ORACLE_SID is test

       - UNIX user is ora8 on both camel and giraffe.

1. Create the file, *install_path*/netbackup/db/altnames/giraffe, on server lion and edit it to contain the name camel.

2. Log in to giraffe as ora8.

3. Set SERVER=lion in *$ORACLE_HOME*/bp.conf.

4. Modify the network tnsnames.ora file to enable RMAN catalog connection.

5. Set the environment variables ORACLE_SID to test and NB_ORA_CLIENT to camel.

6. Make sure the directory exists and has appropriate access permissions.

   The datafiles will be restored to a directory path with the same name they had when backed up.

7. Create an inittest.ora file.

8. Start up the database in a nomount state.

```
SVRMGR> startup mount pfile=$ORACLE_HOME/dbs/inittest.ora
%rman rcvcat rman/rman@rcat
RMAN> set dbid=<dbid of source database on camel>
RMAN> connect target/
RMAN>run {
RMAN>      ALLOCATE CHANNEL CH00 TYPE'SBT_TAPE';
RMAN>      SEND 'NB_ORA_SERV=lion, NB_ORA_CLIENT=camel';
RMAN>      restore controlfile
RMAN>      }


SVRMGR> alter database mount;

%orapwd file=$ORACLE_HOME/dbs/orapwtest password=<oracle>
%rman rcvcat rman/rman@RCVCAT

RMAN>set dbid=<Saved dbID of Source Target>
RMAN>connect target/
RMAN>run {
RMAN>      ALLOCATE CHANNEL CH00 TYPE 'SBT_TAPE';
RMAN>      ALLOCATE CHANNEL CH01 TYPE 'SBT_TAPE';
RMAN>      SEND 'NB_ORA_SERV=lion, NB_ORA_CLIENT=camel';
RMAN>      restore database;
RMAN>      restore archivelog all;
```

```
RMAN>      }

SVRGMR>recover database until cancel using backup controlfile;
```

At this point, apply the archived logs and type cancel when you decide to stop recovery.

Performing Other `rman` Actions

To execute script files for database operations other than backups or restores, it is recommended that you execute the `rman` command directly rather than using NetBackup.

◆ For the `rman` command script syntax and examples, see the *Oracle Recovery Manager User's Guide and Reference*.

Troubleshooting 5

NetBackupNetBackup for Oracle ServerFree Agent provide reports on operations. These reports are useful for finding errors associated with those applications.

◆ NetBackup and NetBackup for Oracle ServerFree Agent Logs

◆ Troubleshooting Procedure

◆ Backup or Restore Errors

NetBackup and NetBackup for Oracle ServerFree Agent Logs

The following describes troubleshooting logs and reports generated by NetBackup products.

Logs

The NetBackup server and client software allow you to set up detailed activity logs for troubleshooting problems that occur outside of either NetBackup for Oracle ServerFree Agent or the Oracle Recovery Manager. See the *NetBackup Troubleshooting Guide for UNIX* for a complete description of activity logs. Also see the *install_path*/netbackup/logs/README.debug file.

> **Note** These logs do not reveal errors that occur during the execution of the Oracle Recovery Manager, unless those errors also affect NetBackup for Oracle ServerFree Agent. Your best sources for Oracle error information are the logs provided by the Oracle.

Enable the NetBackup for Oracle Advanced SF Agent logs by performing the following steps.

1. Create the following directories on the client:

   install_path/netbackup/logs/bphdb
   install_path/netbackup/logs/dbclient
   install_path/netbackup/logs/bpbkar
   install_path/netbackup/logs/tar
   install_path/netbackup/logs/ouline_util

   ```
   % cd install_path/netbackup/logs
   % mkdir bphdb
   % mkdir dbclient
   % mkdir bpbkar
   % mkdir tar
   % mkdir online_util
   ```

2. Set the access permissions to 777 on these log directories.

   ```
   % chmod 777 bphdb
   % chmod 777 dbclient
   % chmod 777 bpbkar
   % chmod 777 tar
   % chmod 777 online_util
   ```

The following sections describe the logs created when you create the log directories. Use a text editor to view the contents of the logs.

bphdb Directory on the Client

The *install_path*/netbackup/logs/bphdb directory contains the following types of logs. These logs are a good starting place to determine what type of error occurred.

obk_stdout.*mmddyy*

> Unless redirected elsewhere, NetBackup places NetBackup for Oracle script output in this file.

obk_stderr.*mmddyy*

> Unless redirected elsewhere, NetBackup places NetBackup for Oracle script errors in this file.

log.*mmddyy*

> bphdb is the NetBackup Database Backup binary. This log contains debugging information for the bphdb process. NetBackup for Oracle ServerFree Agent uses this client process for NetBackup for Oracle script execution. It is invoked when an automatic backup schedule is executed.

dbclient Directory on the Client

The *install_path*/netbackup/logs/dbclient directory contains the following execution log.

log.*mmddyy*

> This log contains debugging information and execution status for the Oracle processes linked to the library provided with NetBackup for Oracle ServerFree Agent.

bpbkar Directory on the Client

The *install_path*/netbackup/logs/bpbkar directory contains the following execution log.

log.*mmddyy*

> This log contains debugging information and execution status for the local host backup processing, which performs the actual data transfer of Oracle datafiles for proxy backups.

ouline_util Directory on the Client

The *install_path*/netbackup/logs/online_util directory contains the following execution log.

log.*mmddyy*

> This log contains debugging information and execution status for the ServerFree backup processes, which perform the actual data transfer of Oracle datafiles for proxy backups.

tar Directory on the Client

The *install_path*/netbackup/logs/tar directory contains the following execution log.

log.*mmddyy*

> This log contains debugging information and execution status for the NetBackup process, which performs the actual data transfer of Oracle datafiles for proxy restores.

Setting the Debug Level

You can control the amount of information written to the debug log in the *install_path*/netbackup/logs/dbclient directory by changing the Database debug level. The higher the value, the more information is logged. In everyday normal operations, the default value of 0 is sufficient. However, VERITAS technical support may ask you to set the value higher when a problem is being analyzed. 5 is the highest possible setting.

❖ To change the Debug Level, enter the following line in the bp.conf file.

> VERBOSE = 5

NetBackup Server Reports

NetBackup provides other reports that are useful in isolating problems. One such report is All Logs Entries on the server. See the *NetBackup System Administrator's Guide* for a description of this and other reports.

Oracle Recovery Manager Utility Logs

The Oracle Recovery Manager uses a command language interpreter, and can be executed in interactive or batch mode. You can specify a log file on the command line to record significant RMAN actions. The syntax is:

```
msglog 'logfile name'
```

Troubleshooting Procedure

Note For brevity, this chapter refers to the `libobk` module as API. Many media manager vendors also call the `libobk` module DMO (Database Module).

To perform this procedure, the following conditions must exist.

The following products are properly installed and configured:

- NetBackup 4.5 ServerFree Agent
- Oracle RDBMS 8i or later
- NetBackup for Oracle ServerFree Agent

Refer to the *NetBackup 4.5 ServerFree Agent System Administrator's Guide* and the *NetBackup 4.5 for EMC Fastrax System Administrator's Guide* for more troubleshooting information.

1. When verifying your installation, ensure that the following two NetBackup for Oracle ServerFree Agent binaries exist (in *install_path*/netbackup/bin):

- `bphdb` resides on the client and is used by both the NetBackup scheduler and the graphical interface to start backups. The main purpose of `bphdb` is to execute an Oracle template or shell script that in turn calls `rman`. Only the RMAN script is required when `rman` is executed from the command line.

- `libobk` is a shared library module containing functions callable by the Oracle Recovery Manager. This library is loaded when Oracle Recovery Manager is started. The actual name of this binary is dependent on the operating system.

    - `libobk.so.1` on Solaris (32-bit)
    - `libobk.so64.1` on Solaris (64-bit)
    - `libobk.sl` on HP-UX (32-bit)
    - `libobk.sl64` on HP-UX (64-bit)

2. Check that both the NetBackup server and client software are working properly. That is, check that normal operating system files can be backed up and restored from the client.

The following NetBackup logs are turned on in `VERBOSE` mode:

- On the client: `dbclient`, `bphdb`, `bpbkar`, `tar`, `ouline_util` and `bpcd`.

- On the master server: `bprd`, `bpsched`, and `bpdbm`.

- On the host with the storage unit: `bpbrm`, and `bptm`.

Note These logs may become very large, especially `bpsched` and `bpdbm`. Ensure that enough free disk space exists in the log directory disk partition.

Backup or Restore Errors

An Oracle Recovery Manager backup error can originate:

◆ On the NetBackup side

An error can be from the API, the NetBackup server or client, or Media Manager.

◆ On the Oracle side

The error can be from the Oracle Recovery Manager, or the target database.

VERITAS suggests that you use the following steps when troubleshooting a failed operation:

1. Check the logs to determine the source of the error.

2. Troubleshoot each stage of the backup or restore.

The following sections describe these steps in detail.

Check the Logs to Determine the Source of the Error

Determine whether the error is from NetBackup or the Oracle Recovery Manager.

If the error came from NetBackup, go to "Troubleshoot Each Stage of the Backup or Restore" on page 115.

Otherwise, have the Oracle database administrator or the NetBackup administrator look at the problem.

The majority of the problems setting up NetBackup and RMAN concern the linking process not working, an error with the RMAN script that is being executed, issues with the NetBackup policy being set up incorrectly, or problems allocating tape devices/media.

> **Note** Ensure that server and user-directed regular file system backups work for the client prior to debugging the Oracle RMAN backups. Try backing up the Oracle binaries.

◆ Always start by running `rman` from the command line rather than having NetBackup scheduler execute a script. These troubleshooting steps assume you are running `rman` from the command line.

◆ If NetBackup and Oracle's `rman` are communicating, you should see a log in *install_path*/netbackup/logs/dbclient. If a log file exists and you are having problems, this usually means a configuration problem within the NetBackup policy or media availability.

```
cd install_path/netbackup/logs/dbclient
ls
  log.123098
```

If you have a log file, then NetBackup and Oracle RMAN are linked correctly.

◆ If NetBackup and Oracle's `rman` are NOT communicating, you will NOT see a log in *install_path*/netbackup/logs/dbclient.

```
cd install_path/netbackup/logs/dbclient
ls
```

Logs will not be created in this directory if the permissions are not set for the Oracle user to write the directory. Full permissions is best, chmod 777.

Verify that the RMAN script works correctly. That is, verify that the RMAN script will back up to `allocate channel c1 type disk` instead of `allocate channel c1 type sbt_tape`.

If RMAN can back up to disk and not to tape, and no log file exists in `dbclient`, then the link did not work correctly.

If you suspect a link failure, refer to "Linking Oracle8 with NetBackup for Oracle on UNIX" on page 28.

Troubleshoot Each Stage of the Backup or Restore

The following explains the sequence of events for an Oracle Recovery Manager initiated action and suggests solutions for problems that can occur at each point in the sequence.

1. `rman` starts.

A backup or restore can be started in any of the following ways:

- Command line from the system prompt such as:

  For Oracle Recovery Manager Backup or Restore:

```
% rman target user/pwd[@TNS alias] \
  rcvcat user/pwd[@TNS alias]\
  cmdfile RMAN script file name
```

where *RMAN script file name* is fully qualified.

- Manually from the administrator interface on the master server.

- Automatically by an automatic backup schedule.

If an error occurs at this point, check the Oracle Recovery Manager log.

2. Oracle Recovery Manager verifies its environment, then issues requests to the API.

Some information such as the NetBackup version, API versions, and trace filename, and NetBackup signal handlers are registered with the Oracle Recovery Manager.

An error at this point is usually due to a problem with client and server communication. Check the messages in the `bprd`, `bpsched`, and `bpcd` logs for clues. Also verify the `bp.conf` entries on the client.

3. Oracle Recovery Manager issues a backup or restore request.

The API gathers necessary parameters, and then sends the `backup` or `restore` request to the NetBackup server. The API waits until both the server and client are ready to transfer data before returning.

Oracle Recovery Manager requests a backup or restore by passing the following to the API:

- Filename it wants NetBackup to use

- Mode (backup or restore)

- Actual name of the Oracle data file

- Information about the target database

The API gathers information from the following:

- Environment

- User's `bp.conf` and *install_path*`/netbackup/bp.conf` files on the client

- Information file from `bphdb` to be used as parameters to the `bprd` process

The API then sends this information to the master server's `bprd` process.

To troubleshoot a problem in this part of the first sequence, examine the *install_path*`/netbackup/logs/dbclient/log.`*mmddyy.*

- If the `bprd` process failed, check the logs for `bprd`, `bpbrm`, and `bpsched`.

A failure at this point is frequently due to bad NetBackup server or Oracle policy configuration parameters:

NetBackup can usually select the correct Oracle policy and schedules but not always when there are several Oracle policies in its database. Try setting the SERVER and POLICY values in the `bp.conf` file on the client or by using environment variables. For example, the following C Shell `setenv` commands specify the Oracle policy, schedule, and server for NetBackup to use:

```
setenv NB_ORA_POLICY policyname
setenv NB_ORA_SCHED schedule
setenv NB_ORA_SERV NetBackup server
```

4. Oracle Recovery Manager issues read or write requests to the API, which then transfers data to or from the NetBackup server.

A failure here is probably due to NetBackup media, network, or timeout errors.

5. Oracle Recovery Manager tells the API to close the session.

The API waits for the server to complete its necessary actions (backup image verification and so on) and then exits.

An error can originate from either NetBackup or Oracle Recovery Manager.

- Oracle Recovery Manager will abort if they encounter an error while reading a data file during the backup (for example, if Oracle blocks are out of sequence). It also aborts if NetBackup sends a bad backup image during the restore.

- NetBackup may return an error code to the API if for some reason it could not complete the backup successfully.

Preventing Timeout Failures on Large Database Restores

Large database restores sometimes fail when multiple restore sessions compete for resources. In this situation, a restore session can be delayed waiting for media or device access. If the delay is too long, the restore session will timeout.

This problem can be resolved by increasing the NetBackup Client Read Timeout setting, which will prevent session timeouts and allow the restores to complete successfully.

Use the NetBackup Administration Console on the server to change the properties of each client that contains a database you may need to restore. The default for the Client Read Timeout setting is 300 seconds (5 minutes). For database agent clients, increase the value significantly to prevent timeout errors, e.g. 30 minutes.

NetBackup for Oracle RMAN Scripts $\quad$ **A**

Example 1, `cold_database_backup.sh`

This example sets the environment and calls rman with the appropriate command file to perform a whole database backup. It is used for both full backups and incremental backups. When a schedule executes, NetBackup sets environment variables that the script uses to perform a backup.

```
#
# -------------------------------------------------------------------------------
#  cold_database_backup.sh
# -------------------------------------------------------------------------------
#  This script uses Recovery Manager to take a cold (consistent) database
#  backup. A cold backup is one where the database is shutdown cleanly before
#  performing the backup. It is assumed that this script will be executed
#  by user root. In order for the Oracle utilities, sqlplus, svrmgrl and rman,
#  to work properly we switch user (su -) to the oracle dba account before
#  each execution. If this script runs under a user account that has Oracle
#  dba privilege, it will be executed using this user's account.
#
#  Note: As written, this script will bring down the target database and
#        open it when done.
# -------------------------------------------------------------------------------


# -------------------------------------------------------------------------------
# Determine the user which is executing this script.
# -------------------------------------------------------------------------------

CUSER=`id |cut -d"(" -f2 | cut -d ")" -f1`

# -------------------------------------------------------------------------------
# Put output in <this file name>.out. Change as desired.
# Note: output directory requires write permission.
# -------------------------------------------------------------------------------

RMAN_LOG_FILE=${0}.out
```

```
# ----------------------------------------------------------------------------
# You may want to delete the output file so that backup information does
# not accumulate.  If not, delete the following lines.
# ----------------------------------------------------------------------------

if [ -f "$RMAN_LOG_FILE" ]
then
rm -f "$RMAN_LOG_FILE"
fi

# ------------------------------------------------------------------
# Initialize the log file.
# ------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
chmod 666 $RMAN_LOG_FILE

# ----------------------------------------------------------------------------
# Log the start of this script.
# ----------------------------------------------------------------------------

echo Script $0 >> $RMAN_LOG_FILE
echo ==== started on `date` ==== >> $RMAN_LOG_FILE
echo >> $RMAN_LOG_FILE

# ----------------------------------------------------------------------------
# Replace /db/oracle/product/ora81, below, with the Oracle home path.
# ----------------------------------------------------------------------------

ORACLE_HOME=/db/oracle/product/ora81
export ORACLE_HOME

# ----------------------------------------------------------------------------
# Replace ora81, below, with the Oracle SID of the target database.
# ----------------------------------------------------------------------------

ORACLE_SID=ora81
export ORACLE_SID

# ----------------------------------------------------------------------------
# Replace ora81, below, with the Oracle DBA user id (account).
# ----------------------------------------------------------------------------

ORACLE_USER=ora81

# ----------------------------------------------------------------------------
```

```
# Set the target connect string.
# Replace "sys/manager", below, with the target connect string.
# This connect string is used by the Oracle utility and rman
# ------------------------------------------------------------------------------

TARGET_CONNECT_STR=sys/manager

# ------------------------------------------------------------------------------
# Set the Oracle utility.
# Replace sqlplus with the appropriate Oracle utility call.
# For example, CMD=$ORACLE_HOME/bin/svrmgrl
# ------------------------------------------------------------------------------
# CMD="$ORACLE_HOME/bin/sqlplus /nolog"
# CMD="$ORACLE_HOME/bin/svrmgrl"

CMD="$ORACLE_HOME/bin/sqlplus /nolog"

# ------------------------------------------------------------------------------
# Relplace the following with the fully qualified path name of the init file
# for your target database.
# ------------------------------------------------------------------------------
TARGET_INIT_FILE=/db/oracle/product/admin/ora81/pfile/initorac81.ora

# ------------------------------------------------------------------------------
# Set the Oracle Recovery Manager name.
# ------------------------------------------------------------------------------

RMAN=$ORACLE_HOME/bin/rman

# ------------------------------------------------------------------------------
# Print out the value of the variables set by this script.
# ------------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
echo    "RMAN: $RMAN" >> $RMAN_LOG_FILE
echo    "CMD: $CMD" >> $RMAN_LOG_FILE
echo    "ORACLE_SID: $ORACLE_SID" >> $RMAN_LOG_FILE
echo    "ORACLE_USER: $ORACLE_USER" >> $RMAN_LOG_FILE
echo    "ORACLE_HOME: $ORACLE_HOME" >> $RMAN_LOG_FILE

# ------------------------------------------------------------------------------
# Print out the value of the variables set by bphdb.
# ------------------------------------------------------------------------------

echo
echo    "NB_ORA_FULL: $NB_ORA_FULL" >> $RMAN_LOG_FILE
```

```
echo    "NB_ORA_INCR: $NB_ORA_INCR" >> $RMAN_LOG_FILE
echo    "NB_ORA_CINC: $NB_ORA_CINC" >> $RMAN_LOG_FILE
echo    "NB_ORA_SERV: $NB_ORA_SERV" >> $RMAN_LOG_FILE
echo    "NB_ORA_POLICY: $NB_ORA_POLICY" >> $RMAN_LOG_FILE

# ----------------------------------------------------------------------------
# Shutdown the target database in immediate priority.
# This lets current calls to the database complete, but prevents further
# logons or calls.
#
# The shutdown-startup logic of this script can be commented out if you know
# that the database will always be mounted and in a consistent state before
# this script is executed.
#
# CMD_STR is set up with the proper sequence of commands to initiate the
# Oracle utility, connect to the database and perform the shutdown.
#
# The current user is checked and if it is root a su to the oracle user
# is performed and then the CMD_STR is executed.  This will be performed
# for all scheduled backups.  If the current user is not root, a user
# directed backup is assumed and the CMD_STR is just executed.
# ----------------------------------------------------------------------------

CMD_STR="
ORACLE_HOME=$ORACLE_HOME
export ORACLE_HOME
ORACLE_SID=$ORACLE_SID
export ORACLE_SID
$CMD << EOF
connect $TARGET_CONNECT_STR as sysdba
shutdown immediate
exit
EOF
"
# Initiate the command string

if [ "$CUSER" = "root" ]
then
    su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
else
    /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
fi


# ----------------------------------------------------------------------------
# Now we know that the database is cleanly closed and is ready for a
```

```
# cold backup. RMAN requires that the database be started and mounted
# to perform a backup.
#
# CMD_STR is set up with the proper sequence of commands to initiate the
# Oracle utility, connect to the database, startup the database associated
# to the specified pfile in a mounted mode.
#
# The current user is checked and if it is root a su to the oracle user
# is performed and then the CMD_STR is executed.  This will be performed
# for all scheduled backups.  If the current user is not root, a user
# directed backup is assumed and the CMD_STR is just executed.
# ----------------------------------------------------------------------------

CMD_STR="
ORACLE_HOME=$ORACLE_HOME
export ORACLE_HOME
ORACLE_SID=$ORACLE_SID
export ORACLE_SID
$CMD << EOF
connect $TARGET_CONNECT_STR as sysdba
startup mount pfile=$TARGET_INIT_FILE
exit
EOF
"
# Initiate the command string

if [ "$CUSER" = "root" ]
then
    su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
else
    /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
fi

echo >> $RMAN_LOG_FILE

# ----------------------------------------------------------------------------
# If this script is executed from a NetBackup schedule, NetBackup
# sets an NB_ORA environment variable based on the schedule type.
# The NB_ORA variable is then used to dynamically set BACKUP_TYPE
# For example, when:
#     schedule type is                 BACKUP_TYPE is
#     ----------------                 --------------
# Automatic Full                       INCREMENTAL LEVEL=0
# Automatic Differential Incremental   INCREMENTAL LEVEL=1
# Automatic Cumulative Incremental     INCREMENTAL LEVEL=1 CUMULATIVE
#
```

```
# For user initiated backups, BACKUP_TYPE defaults to incremental
# level 0 (full).  To change the default for a user initiated
# backup to incremental or incremental cumulative, uncomment
# one of the following two lines.
# BACKUP_TYPE="INCREMENTAL LEVEL=1"
# BACKUP_TYPE="INCREMENTAL LEVEL=1 CUMULATIVE"
#
# Note that we use incremental level 0 to specify full backups.
# That is because, although they are identical in content, only
# the incremental level 0 backup can have incremental backups of
# level > 0 applied to it.
# ---------------------------------------------------------------------------

if [ "$NB_ORA_FULL" = "1" ]
then
echo "Full backup requested" >> $RMAN_LOG_FILE
BACKUP_TYPE="INCREMENTAL LEVEL=0"

elif [ "$NB_ORA_INCR" = "1" ]
then
echo "Differential incremental backup requested" >> $RMAN_LOG_FILE
BACKUP_TYPE="INCREMENTAL LEVEL=1"

elif [ "$NB_ORA_CINC" = "1" ]
then
echo "Cumulative incremental backup requested" >> $RMAN_LOG_FILE
BACKUP_TYPE="INCREMENTAL LEVEL=1 CUMULATIVE"

elif [ "$BACKUP_TYPE" = "" ]
then
echo "Default - Full backup requested" >> $RMAN_LOG_FILE
BACKUP_TYPE="INCREMENTAL LEVEL=0"
fi

# ---------------------------------------------------------------------------
# Call Recovery Manager to initiate the backup. This example does not use a
# Recovery Catalog. If you choose to use one, replace the option 'nocatalog'
# from the rman command line below with the
# 'rcvcat <userid>/<passwd>@<tns alias>' statement.
#
# Note: Any environment variables needed at run time by RMAN
#       must be set and exported within the switch user (su) command.
# ---------------------------------------------------------------------------

CMD_STR="
ORACLE_HOME=$ORACLE_HOME
```

```
            export ORACLE_HOME
            ORACLE_SID=$ORACLE_SID
            export ORACLE_SID
            $RMAN target $TARGET_CONNECT_STR nocatalog msglog $RMAN_LOG_FILE append << EOF
            RUN {
            ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
            ALLOCATE CHANNEL ch01 TYPE 'SBT_TAPE';
            BACKUP
                $BACKUP_TYPE
                FORMAT 'bk_u%u_s%s_p%p_t%t'
                DATABASE;
                sql 'alter database open';
            RELEASE CHANNEL ch00;
            RELEASE CHANNEL ch01;
            }
            EOF
            "
            # Initiate the command string

            if [ "$CUSER" = "root" ]
            then
                su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
                RSTAT=$?
            else
                /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
                RSTAT=$?
            fi

            # -----------------------------------------------------------------------------
            # Log the completion of this script.
            # -----------------------------------------------------------------------------

            if [ "$RSTAT" = "0" ]
            then
                LOGMSG="ended successfully"
            else
                LOGMSG="ended in error"
            fi

            echo >> $RMAN_LOG_FILE
            echo Script $0 >> $RMAN_LOG_FILE
            echo ==== $LOGMSG on `date` ==== >> $RMAN_LOG_FILE
            echo >> $RMAN_LOG_FILE

            exit $RSTAT
```

Example 2, `hot_database_backup.sh`

This example sets up the environment and calls `rman` with the appropriate command file, to perform a whole database backup. It is used for both full backups and incremental backups. When a schedule executes, NetBackup sets environment variables that the script uses to perform the backup.

With the proper schedules this script could be used to automatically execute a backup every week on Friday night and a incremental backup each night for the rest of the week.

```
#
# ------------------------------------------------------------------------------
#  hot_database_backup.sh
# ------------------------------------------------------------------------------
#  This script uses Recovery Manager to take a hot (inconsistent) database
#  backup. A hot backup is inconsistent because portions of the database are
#  being modified and written to the disk while the backup is progressing.
#  You must run your database in ARCHIVELOG mode to make hot backups. It is
#  assumed that this script will be executed by user root. In order for RMAN
#  to work properly we switch user (su -) to the oracle dba account before
#  execution. If this script runs under a user account that has Oracle dba
#  privilege, it will be executed using this user's account.
# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------------
# Determine the user which is executing this script.
# ------------------------------------------------------------------------------

CUSER=`id |cut -d"(" -f2 | cut -d ")" -f1`

# ------------------------------------------------------------------------------
# Put output in <this file name>.out. Change as desired.
# Note: output directory requires write permission.
# ------------------------------------------------------------------------------

RMAN_LOG_FILE=${0}.out

# ------------------------------------------------------------------------------
# You may want to delete the output file so that backup information does
# not accumulate.  If not, delete the following lines.
# ------------------------------------------------------------------------------

if [ -f "$RMAN_LOG_FILE" ]
then
rm -f "$RMAN_LOG_FILE"
fi
```

```
# --------------------------------------------------------------------
# Initialize the log file.
# --------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
chmod 666 $RMAN_LOG_FILE


# ------------------------------------------------------------------------------
# Log the start of this script.
# ------------------------------------------------------------------------------

echo Script $0 >> $RMAN_LOG_FILE
echo ==== started on `date` ==== >> $RMAN_LOG_FILE
echo >> $RMAN_LOG_FILE


# ------------------------------------------------------------------------------
# Replace /db/oracle/product/ora81, below, with the Oracle home path.
# ------------------------------------------------------------------------------

ORACLE_HOME=/db/oracle/product/ora81
export ORACLE_HOME


# ------------------------------------------------------------------------------
# Replace ora81, below, with the Oracle SID of the target database.
# ------------------------------------------------------------------------------

ORACLE_SID=ora81
export ORACLE_SID


# ------------------------------------------------------------------------------
# Replace ora81, below, with the Oracle DBA user id (account).
# ------------------------------------------------------------------------------

ORACLE_USER=ora81


# ------------------------------------------------------------------------------
# Set the target connect string.
# Replace "sys/manager", below, with the target connect string.
# ------------------------------------------------------------------------------

TARGET_CONNECT_STR=sys/manager


# ------------------------------------------------------------------------------
# Set the Oracle Recovery Manager name.
# ------------------------------------------------------------------------------
```

```
RMAN=$ORACLE_HOME/bin/rman

# ----------------------------------------------------------------------------
# Print out the value of the variables set by this script.
# ----------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
echo    "RMAN: $RMAN" >> $RMAN_LOG_FILE
echo    "ORACLE_SID: $ORACLE_SID" >> $RMAN_LOG_FILE
echo    "ORACLE_USER: $ORACLE_USER" >> $RMAN_LOG_FILE
echo    "ORACLE_HOME: $ORACLE_HOME" >> $RMAN_LOG_FILE

# ----------------------------------------------------------------------------
# Print out the value of the variables set by bphdb.
# ----------------------------------------------------------------------------

echo  >> $RMAN_LOG_FILE
echo    "NB_ORA_FULL: $NB_ORA_FULL" >> $RMAN_LOG_FILE
echo    "NB_ORA_INCR: $NB_ORA_INCR" >> $RMAN_LOG_FILE
echo    "NB_ORA_CINC: $NB_ORA_CINC" >> $RMAN_LOG_FILE
echo    "NB_ORA_SERV: $NB_ORA_SERV" >> $RMAN_LOG_FILE
echo    "NB_ORA_POLICY: $NB_ORA_POLICY" >> $RMAN_LOG_FILE

# ----------------------------------------------------------------------------
# NOTE: This script assumes that the database is properly opened. If desired,
# this would be the place to verify that.
# ----------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
# ----------------------------------------------------------------------------
# If this script is executed from a NetBackup schedule, NetBackup
# sets an NB_ORA environment variable based on the schedule type.
# The NB_ORA variable is then used to dynamically set BACKUP_TYPE
# For example, when:
#     schedule type is                 BACKUP_TYPE is
#     ---------------                  -------------
# Automatic Full                       INCREMENTAL LEVEL=0
# Automatic Differential Incremental   INCREMENTAL LEVEL=1
# Automatic Cumulative Incremental     INCREMENTAL LEVEL=1 CUMULATIVE
#
# For user initiated backups, BACKUP_TYPE defaults to incremental
# level 0 (full).  To change the default for a user initiated
# backup to incremental or incremental cumulative, uncomment
# one of the following two lines.
# BACKUP_TYPE="INCREMENTAL LEVEL=1"
```

```
                # BACKUP_TYPE="INCREMENTAL LEVEL=1 CUMULATIVE"
                #
                # Note that we use incremental level 0 to specify full backups.
                # That is because, although they are identical in content, only
                # the incremental level 0 backup can have incremental backups of
                # level > 0 applied to it.
                # ---------------------------------------------------------------------------

                if [ "$NB_ORA_FULL" = "1" ]
                then
                        echo "Full backup requested" >> $RMAN_LOG_FILE
                        BACKUP_TYPE="INCREMENTAL LEVEL=0"

                elif [ "$NB_ORA_INCR" = "1" ]
                then
                        echo "Differential incremental backup requested" >> $RMAN_LOG_FILE
                        BACKUP_TYPE="INCREMENTAL LEVEL=1"

                elif [ "$NB_ORA_CINC" = "1" ]
                then
                        echo "Cumulative incremental backup requested" >> $RMAN_LOG_FILE
                        BACKUP_TYPE="INCREMENTAL LEVEL=1 CUMULATIVE"

                elif [ "$BACKUP_TYPE" = "" ]
                then
                        echo "Default - Full backup requested" >> $RMAN_LOG_FILE
                        BACKUP_TYPE="INCREMENTAL LEVEL=0"
                fi


                # ---------------------------------------------------------------------------
                # Call Recovery Manager to initiate the backup. This example does not use a
                # Recovery Catalog. If you choose to use one, replace the option 'nocatalog'
                # from the rman command line below with the
                # 'rcvcat <userid>/<passwd>@<tns alias>' statement.
                #
                # Note: Any environment variables needed at run time by RMAN
                #       must be set and exported within the switch user (su) command.
                # ---------------------------------------------------------------------------
                #  Backs up the whole database.  This backup is part of the incremental
                #  strategy (this means it can have incremental backups of levels > 0
                #  applied to it).
                #
                #  We do not need to explicitly request the control file to be included
                #  in this backup, as it is automatically included each time file 1 of
                #  the system tablespace is backed up (the inference: as it is a whole
```

```
# database backup, file 1 of the system tablespace will be backed up,
# hence the controlfile will also be included automatically).
#
# Typically, a level 0 backup would be done at least once a week.
#
# The scenario assumes:
#    o you are backing your database up to two tape drives
#    o you want each backup set to include a maximum of 5 files
#    o you wish to include offline datafiles, and read-only tablespaces,
#       in the backup
#    o you want the backup to continue if any files are inaccessible.
#    o you are not using a Recovery Catalog
#    o you are explicitly backing up the control file.  Since you are
#      specifying nocatalog, the controlfile backup that occurs
#      automatically as the result of backing up the system file is
#      not sufficient; it will not contain records for the backup that
#      is currently in progress.
#    o you want to archive the current log, back up all the
#      archive logs using two channels, putting a maximum of 20 logs
#      in a backup set, and deleting them once the backup is complete.
#
# Note that the format string is constructed to guarantee uniqueness and
# to enhance NetBackup for Oracle backup and restore performance.
#
# ------------------------------------------------------------------------------

CMD_STR="
ORACLE_HOME=$ORACLE_HOME
export ORACLE_HOME
ORACLE_SID=$ORACLE_SID
export ORACLE_SID
$RMAN target $TARGET_CONNECT_STR nocatalog msglog $RMAN_LOG_FILE append << EOF
RUN {
ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
ALLOCATE CHANNEL ch01 TYPE 'SBT_TAPE';
BACKUP
    $BACKUP_TYPE
    SKIP INACCESSIBLE
    TAG hot_db_bk_level0
    FILESPERSET 5
    # recommended format
    FORMAT 'bk_%s_%p_%t'
    DATABASE;
    sql 'alter system archive log current';
RELEASE CHANNEL ch00;
RELEASE CHANNEL ch01;
```

```
# backup all archive logs
ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
ALLOCATE CHANNEL ch01 TYPE 'SBT_TAPE';
BACKUP
   filesperset 20
   FORMAT 'al_%s_%p_%t'
   ARCHIVELOG ALL DELETE INPUT;
RELEASE CHANNEL ch00;
RELEASE CHANNEL ch01;
#
# Note: During the process of backing up the database, RMAN also backs up the
# control file.  This version of the control file does not contain the
# information about the current backup because "nocatalog" has been specified.
# Too include the information about the current backup, the control file should
# be backed up as the last step of the RMAN section.  This step would not be
# necessary if we were using a recovery catalog.
#
ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
BACKUP
    # recommended format
    FORMAT 'cntrl_%s_%p_%t'
    CURRENT CONTROLFILE;
RELEASE CHANNEL ch00;
}
EOF
"
# Initiate the command string

if [ "$CUSER" = "root" ]
then
    su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
    RSTAT=$?
else
    /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
    RSTAT=$?
fi

# ----------------------------------------------------------------------------
# Log the completion of this script.
# ----------------------------------------------------------------------------

if [ "$RSTAT" = "0" ]
then
    LOGMSG="ended successfully"
else
    LOGMSG="ended in error"
```

```
        fi

        echo >> $RMAN_LOG_FILE
        echo Script $0 >> $RMAN_LOG_FILE
        echo ==== $LOGMSG on `date` ==== >> $RMAN_LOG_FILE
        echo >> $RMAN_LOG_FILE

        exit $RSTAT
```

Example 3, `cold_duplex_database_backup_full.sh`

This example sets up the environment and calls `rman` to execute a command file that will make two copies of a cold (consistent) database backup.

```
        #
        # ----------------------------------------------------------------------------
        #  cold_duplex_database_backup_full.sh
        # ----------------------------------------------------------------------------
        # This script uses Recovery Manager to make multiple copies of a cold
        # (consistent) database backup. A cold backup is one where the database
        # is shutdown cleanly before performing the backup. It is assumed that
        # this script will be executed by user root. In order for RMAN to work
        # properly we switch user (su -) to the oracle dba account. If this script
        # runs under a user account that has Oracle dba privilege, it will be
        # executed using this user's account.
        #
        # Note that you must enable (set to TRUE in the init.ora file) the
        # BACKUP_TAPE_IO_SLAVES initialization parameter to perform duplexed
        # backups. RMAN will configure as many slaves as needed for the number
        # of backup copies you request. For more information on
        # BACKUP_TAPE_IO_SLAVES, see the Oracle Reference.
        # ----------------------------------------------------------------------------

        # ----------------------------------------------------------------------------
        # Determine the user which is executing this script.
        # ----------------------------------------------------------------------------

        CUSER=`id |cut -d"(" -f2 | cut -d ")" -f1`

        # ----------------------------------------------------------------------------
        # Put output in <this file name>.out. Change as desired.
        # Note: output directory requires write permission.
        # ----------------------------------------------------------------------------

        RMAN_LOG_FILE=${0}.out
```

```
# --------------------------------------------------------------------------
# You may want to delete the output file so that backup information does
# not accumulate.  If not, delete the following lines.
# --------------------------------------------------------------------------

if [ -f "$RMAN_LOG_FILE" ]
then
rm -f "$RMAN_LOG_FILE"
fi

# -------------------------------------------------------------------
# Initialize the log file.
# -------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
chmod 666 $RMAN_LOG_FILE

# --------------------------------------------------------------------------
# Log the start of this script.
# --------------------------------------------------------------------------

echo Script $0 >> $RMAN_LOG_FILE
echo ==== started on `date` ==== >> $RMAN_LOG_FILE
echo >> $RMAN_LOG_FILE

# --------------------------------------------------------------------------
# Replace /db/oracle/product/ora81, below, with the Oracle home path.
# --------------------------------------------------------------------------

ORACLE_HOME=/db/oracle/product/ora81
export ORACLE_HOME

# --------------------------------------------------------------------------
# Replace ora81, below, with the Oracle SID of the target database.
# --------------------------------------------------------------------------

ORACLE_SID=ora81
export ORACLE_SID

# --------------------------------------------------------------------------
# Replace ora81, below, with the Oracle DBA user id (account).
# --------------------------------------------------------------------------

ORACLE_USER=ora81

# --------------------------------------------------------------------------
```

```
# Replace the following with the fully qualified path name of the init file
# for your target database.
# ------------------------------------------------------------------------------

TARGET_INIT_FILE=/db/oracle/product/admin/ora81/pfile/initorac81.ora

# ------------------------------------------------------------------------------
# Replace productionDB, below, with the target database TNS alias (service)
# name from the tnsnames.ora file.
# ------------------------------------------------------------------------------

TARGET_TNS=productionDB
export TARGET_TNS

# ------------------------------------------------------------------------------
# Replace rcvcatDB, below, with the recovery catalog database TNS alias
# (service) name from the tnsnames.ora file.
# ------------------------------------------------------------------------------

RCVCAT_TNS=rcvcatDB
export RCVCAT_TNS

# ------------------------------------------------------------------------------
# Set the rcvcat connect string.
# Replace "rman/rman", below, with the rcvcat connect string.
# ------------------------------------------------------------------------------

RCVCAT_CONNECT_STR=rman/rman

# ------------------------------------------------------------------------------
# Set the Oracle utility.
# Replace sqlplus with the appropriate Oracle utility call.
# For example, CMD=$ORACLE_HOME/bin/svrmgrl
# ------------------------------------------------------------------------------
# CMD="$ORACLE_HOME/bin/sqlplus /nolog"
# CMD="$ORACLE_HOME/bin/svrmgrl"

CMD="$ORACLE_HOME/bin/sqlplus /nolog"

# ------------------------------------------------------------------------------
# Set the target connect string.
# Replace "sys/manager", below, with the target connect string.
# ------------------------------------------------------------------------------

TARGET_CONNECT_STR=sys/manager
```

```
# -----------------------------------------------------------------------------
# Set the Oracle Recovery Manager name.
# -----------------------------------------------------------------------------

RMAN=$ORACLE_HOME/bin/rman

# -----------------------------------------------------------------------------
# Print out the value of the variables set by this script.
# -----------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
echo    "RMAN: $RMAN" >> $RMAN_LOG_FILE
echo    "CMD: $CMD" >> $RMAN_LOG_FILE
echo    "TARGET_TNS: $TARGET_TNS" >> $RMAN_LOG_FILE
echo    "RCVCAT_TNS: $RCVCAT_TNS" >> $RMAN_LOG_FILE
echo    "ORACLE_USER: $ORACLE_USER" >> $RMAN_LOG_FILE
echo    "ORACLE_HOME: $ORACLE_HOME" >> $RMAN_LOG_FILE

echo >> $RMAN_LOG_FILE

# -----------------------------------------------------------------------------
# Shutdown the target database in immediate priority.
# This lets current calls to the database complete, but prevents further
# logons or calls.
#
# The shutdown-startup logic of this script can be commented out if you know
# that the database will always be mounted and in a consistent state before
# this script is executed.
#
# CMD_STR is set up with the proper sequence of commands to initiate the
# Oracle utility, connect to the database and perform the shutdown.
#
# The current user is checked and if it is root a su to the oracle user
# is performed and then the CMD_STR is executed.  This will be performed
# for all scheduled backups.  If the current user is not root, a user
# directed backup is assumed and the CMD_STR is just executed.
# -----------------------------------------------------------------------------

CMD_STR="
ORACLE_HOME=$ORACLE_HOME
export ORACLE_HOME
ORACLE_SID=$ORACLE_SID
export ORACLE_SID
$CMD << EOF
connect $TARGET_CONNECT_STR@$TARGET_TNS as sysdba
shutdown immediate
```

```
exit
EOF
"
# Initiate the command string

if [ "$CUSER" = "root" ]
then
    su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
else
    /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
fi


echo >> $RMAN_LOG_FILE


# ----------------------------------------------------------------------------
# Now we know that the database is cleanly closed and is ready for a
# cold backup. RMAN requires that the database be started and mounted
# to perform a backup.
#
# CMD_STR is set up with the proper sequence of commands to initiate the
# Oracle utility, connect to the database, startup the database associated
# to the specified pfile in a mounted mode.
#
# The current user is checked and if it is root a su to the oracle user
# is performed and then the CMD_STR is executed.  This will be performed
# for all scheduled backups.  If the current user is not root, a user
# directed backup is assumed and the CMD_STR is just executed.
# ----------------------------------------------------------------------------

CMD_STR="
ORACLE_HOME=$ORACLE_HOME
export ORACLE_HOME
ORACLE_SID=$ORACLE_SID
export ORACLE_SID
$CMD << EOF
connect $TARGET_CONNECT_STR@$TARGET_TNS as sysdba
startup mount pfile=$TARGET_INIT_FILE
exit
EOF
"
# Initiate the command string

if [ "$CUSER" = "root" ]
then
```

```
        su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
    else
        /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
    fi


    echo >> $RMAN_LOG_FILE

    # --------------------------------------------------------------------------
    # Call Recovery Manager to initiate the backup. This example uses a
    # Recovery Catalog. If you choose not to use one, replace the option
    # 'rcvcat <userid>/<passwd>@<tns alias>' with the 'nocatalog' option.
    #
    # Note: Any environment variables needed at run time by RMAN
    #       must be set and exported within the switch user (su) command.
    #
    #  NOTE: THIS SCRIPT CONTAINS ORACLE 8.1.X OR GREATER COMMANDS THAT
    #        WILL NOT WORK WITH ORACLE 8.0.X
    #
    #  This script performs a full database backup, creating two identical
    #  copies. A full backup is a non-incremental backup that contain all blocks
    #  of the datafiles. Since it is a non-incremental backup it can not have
    #  incremental backups applied to it.
    #
    #  We do not need to explicitly request the control file to be included
    #  in this backup, as it is automatically included each time file 1 of
    #  the system tablespace is backed up (the inference: as it is a full
    #  database backup, file 1 of the system tablespace will be backed up,
    #  hence the controlfile will also be included automatically).
    #
    #  The scenario assumes:
    #      o you are using Oracle RMAN version 8.1.x or greater.
    #      o you want to perform a non-incremental consistent (cold) full backup
    #      o you want to use the set duplex command to create two (2)
    #        copies of the backup set
    #      o you have enabled (set to TRUE) the BACKUP_TAPE_IO_SLAVES
    #        initialization parameter in the init.ora file.
    #      o you are backing up your database to two tape drives (one for
    #        each copy, as identified by an Application Backup schedule)
    #      o you wish to include offline datafiles, and read-only tablespaces,
    #        in the backup
    #      o you want the backup to terminate if any files are not
    #        accessible
    #      o you want to open the database after the backup completes
    #
    #  Note that the format string is constructed to guarantee uniqueness and
    #  to enhance NetBackup for Oracle backup and restore performance.
```

```
#
# ----------------------------------------------------------------------------


CMD_STR="
ORACLE_HOME=$ORACLE_HOME
export ORACLE_HOME
ORACLE_SID=$ORACLE_SID
export ORACLE_SID

$RMAN           target          $TARGET_CONNECT_STR@$TARGET_TNS          rcvcat
  $RCVCAT_CONNECT_STR@$RCVCAT_TNS msglog $RMAN_LOG_FILE append << EOF

RUN {
set duplex=ON;
ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
ALLOCATE CHANNEL ch01 TYPE 'SBT_TAPE';
send 'CPF1_POLICY=ColdDbFullBk_duplex, CPF1_SCHED=user_cp1';
send 'CPF2_POLICY=ColdDbFullBk_duplex, CPF2_SCHED=user_cp2';
BACKUP
    FULL
    FORMAT 'bk_%U_%t'
    DATABASE;
    sql 'alter database open';
RELEASE CHANNEL ch00;
RELEASE CHANNEL ch01;
}
EOF
"
# Initiate the command string

if [ "$CUSER" = "root" ]
then
    su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
    RSTAT=$?
else
    /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
    RSTAT=$?
fi


# ----------------------------------------------------------------------------
# Log the completion of this script.
# ----------------------------------------------------------------------------

if [ "$RSTAT" = "0" ]
then
```

```
        LOGMSG="ended successfully"
    else
        LOGMSG="ended in error"
    fi

    echo >> $RMAN_LOG_FILE
    echo Script $0 >> $RMAN_LOG_FILE
    echo ==== $LOGMSG on `date` ==== >> $RMAN_LOG_FILE
    echo >> $RMAN_LOG_FILE

    exit $RSTAT
```

As show in "Example 3, cold_duplex_database_backup_full.sh" on page 132, Oracle8i and later RMAN provides an API that allows you to make up to four backup sets simultaneously, each an exact duplicate of the others. Using NetBackup for Oracle ServerFree Agent, you can back up each copy to a different tape to protect against disaster, media damage, or human error. You must use the `set duplex` and the `send` commands to take advantage of this feature.

The `set duplex` command specifies the number of copies of each backup piece that should be created. The `set duplex` command affects all channels allocated after issuing the command and is in effect until explicitly disabled or changed during the session. You cannot issue the `set duplex` command after allocating a channel.

The command syntax is:

```
set duplex = {ON | OFF | 1 | 2 | 3 | 4}
```

By default, duplex is `OFF` (a single backup set is produced). If you specify `ON`, two identical backup sets are produced.

Each output file must have a unique name. You should use the `%U` format specifier to satisfy this restriction. `%U` is equivalent to `%u_%p_%c` and it guarantees the uniqueness of the backup set name in all circumstances.

Note that you must enable the BACKUP_TAPE_IO_SLAVES initialization parameter to perform duplexed backups. RMAN will configure as many media as needed for the number of backup copies you request. For more information on BACKUP_TAPE_IO_SLAVES, see the Oracle Reference.

Use the `send` command to specify the policy and/or schedule to use with each backup. Because NetBackup for Oracle ServerFree Agent uses the policy or schedule to determine what media to use, this information is required for each copy, or an error will occur.

The command syntax is:

```
send 'keyword=value [, keyword=value,...]';
```

The keywords to specify policy are: CPF1_POLICY, CPF2_POLICY, CPF3_POLICY, and CPF4_POLICY, which specify the backup policy for duplexed file 1 through duplexed file 4.

The keywords used to specify schedule are: CPF1_SCHED, CPF2_SCHED, CPF3_SCHED, and CPF4_SCHED, which specify the Application Backup schedule for duplexed file 1 through duplexed file 4.

Example 4, `hot_database_backup_proxy.sh`

This example sets the environment and calls `rman` with the appropriate command file to perform a whole database proxy backup.

When a schedule executes, NetBackup sets environment variables that can be used by NetBackup for Oracle ServerFree Agent.

The script shows how to use the RMAN `send` command to pass the following variables with vendor-specific quoted string:

NB_ORA_POLICY
> Name of the Oracle policy to be used for non-proxy and proxy backups.

NB_ORA_PC_SCHED
> Name of the Oracle schedule to be used for proxy backup.

NB_ORA_PC_BTYPE
> This variable is set to FI if the frozen image attribute is specified for the Oracle policy.

NB_ORA_PC_STREAMS
> Specifies a number of parallel backup streams to be started for each file system in the proxy backup session.

NB_ORA_PC_FILESYSTEMS
> Specifies a number of file systems to be backed up in parallel. The default is to start backups of all file systems in parallel.

Refer to "Multiple Stream Configuration Example" on page 64.

```
#
# ----------------------------------------------------------------------------
#  hot_database_backup_proxy.sh
# ----------------------------------------------------------------------------
#  This script uses Recovery Manager to take a hot (inconsistent) database
#  backup. A hot backup is inconsistent because portions of the database are
#  being modified and written to the disk while the backup is progressing.
#  You must run your database in ARCHIVELOG mode to make hot backups. It is
#  assumed that this script will be executed by user root. In order for RMAN
#  to work properly we switch user (su -) to the oracle dba account before
```

```
#  execution. If this script runs under a user account that has Oracle dba
#  privilege, it will be executed using this user's account.
# ------------------------------------------------------------------------------


# ------------------------------------------------------------------------------
# Determine the user which is executing this script.
# ------------------------------------------------------------------------------


CUSER=`id |cut -d"(" -f2 | cut -d ")" -f1`


# ------------------------------------------------------------------------------
# Put output in <this file name>.out. Change as desired.
# Note: output directory requires write permission.
# ------------------------------------------------------------------------------


RMAN_LOG_FILE=${0}.out


# ------------------------------------------------------------------------------
# You may want to delete the output file so that backup information does
# not accumulate.  If not, delete the following lines.
# ------------------------------------------------------------------------------


if [ -f "$RMAN_LOG_FILE" ]
then
rm -f "$RMAN_LOG_FILE"
fi


# ------------------------------------------------------------------
# Initialize the log file.
# ------------------------------------------------------------------


echo >> $RMAN_LOG_FILE
chmod 666 $RMAN_LOG_FILE


# ------------------------------------------------------------------------------
# Log the start of this script.
# ------------------------------------------------------------------------------


echo Script $0 >> $RMAN_LOG_FILE
echo ==== started on `date` ==== >> $RMAN_LOG_FILE
echo >> $RMAN_LOG_FILE


# ------------------------------------------------------------------------------
# Replace /db/oracle/product/ora81, below, with the Oracle home path.
# ------------------------------------------------------------------------------
```

```
ORACLE_HOME=/db/oracle/product/ora81

# ------------------------------------------------------------------------------
# Replace ora81, below, with the Oracle SID of the target database.
# ------------------------------------------------------------------------------

ORACLE_SID=ora81
export ORACLE_SID

# ------------------------------------------------------------------------------
# Replace ora81, below, with the Oracle DBA user id (account).
# ------------------------------------------------------------------------------

ORACLE_USER=ora81

# ------------------------------------------------------------------------------
# Set the target connect string.
# Replace "sys/manager", below, with the target connect string.
# ------------------------------------------------------------------------------

TARGET_CONNECT_STR=sys/manager

# ------------------------------------------------------------------------------
# Set the Oracle Recovery Manager name.
# ------------------------------------------------------------------------------

RMAN=$ORACLE_HOME/bin/rman

# ------------------------------------------------------------------------------
# Replace the values of the following variables:
#
#        NB_ORA_PC_FILESYSTEMS - specifies this number of file systems to be
#                          backed up in parallel.  The default is the number
#                          of target file systems.  If you do not want to use
#                          the default, you will need to pass this variable
#                          using the send command.
#
#        NB_ORA_PC_STREAMS - specifies this number of parallel backup streams to
#                          start within each file system.
# ------------------------------------------------------------------------------

#NB_ORA_PC_FILESYSTEMS=1
NB_ORA_PC_STREAMS=3

# ------------------------------------------------------------------------------
# Print out the value of the variables set by this script.
```

```
# ------------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
echo    "RMAN: $RMAN" >> $RMAN_LOG_FILE
echo    "ORACLE_SID: $ORACLE_SID" >> $RMAN_LOG_FILE
echo    "ORACLE_USER: $ORACLE_USER" >> $RMAN_LOG_FILE
echo    "ORACLE_HOME: $ORACLE_HOME" >> $RMAN_LOG_FILE
echo    "NB_ORA_SCRIPTS: $NB_ORA_SCRIPTS" >> $RMAN_LOG_FILE
echo    "NB_ORA_PC_STREAMS: $NB_ORA_PC_STREAMS" >> $RMAN_LOG_FILE
echo    "NB_ORA_PC_FILESYSTEMS: $NB_ORA_PC_FILESYSTEMS" >> $RMAN_LOG_FILE

# ------------------------------------------------------------------------------
# Print out the value of the variables passed from the server by bphdb.
# ------------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
echo    "NB_ORA_SERV: $NB_ORA_SERV" >> $RMAN_LOG_FILE
echo    "NB_ORA_POLICY: $NB_ORA_POLICY" >> $RMAN_LOG_FILE
echo    "NB_ORA_PC_SCHED: $NB_ORA_PC_SCHED" >> $RMAN_LOG_FILE
echo    "NB_ORA_PC_BTYPE: $NB_ORA_PC_BTYPE" >> $RMAN_LOG_FILE

# ------------------------------------------------------------------------------
# This script assumes that the database is properly opened. If desired,
# this would be the place to verify that.
# ------------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE

# ------------------------------------------------------------------------------
# Call Recovery Manager to initiate the backup. This example does not use a
# Recovery Catalog. If you choose to use one, replace the option 'nocatalog'
# from the rman command line below and with the
# 'rcvcat <userid>/<passwd>@<tns alias>' statement.
#
# Use the send command on the command line to send NBU policy, schedule name
# and the number of the backup streams to be used in the database proxy backup.
# ------------------------------------------------------------------------------
#
#  Backs up the whole database using proxy copy feature.
#
#  For proxy backups the number of parallel jobs is controlled by
#  two variables, NB_ORA_PC_FILESYSTEMS and NB_ORA_STREAMS, specified
#  earlier in this example.
#
#  You do not need to explicitly request the control file to be included
#  in this backup, as it is automatically included each time file 1 of
```

```
#  the system tablespace is backed up (the inference: as it is a whole
#  database backup, file 1 of the system tablespace will be backed up,
#  hence the controlfile will also be included automatically).
#
#  Note that the controlfiles and archivelogs are not backed up using proxy
#  copy method. Rman will initiate non-proxy copy sessions to backup the
#  controlfile and archivelogs.
#
#
#  The scenario assumes:
#      o you want to backup datafiles using proxy copy method.
#      o you wish to include offline datafiles, and read-only tablespaces,
#        in the backup
#      o you want the backup to continue if any files are inaccessible.
#      o you are not using a Recovery Catalog
#      o you are explicitly backing up the control file.  Since you are
#        specifying nocatalog, the controlfile backup that occurs
#        automatically as the result of backing up the system file is
#        not sufficient; it will not contain records for the backup that
#        is currently in progress.
#      o you want to archive the current log, back up all the
#        archive logs using two channels, putting a maximum of 20 logs
#        in a backup set, and deleting them once the backup is complete.
#
#  Note that the format string is constructed to guarantee uniqueness and
#  to enhance NetBackup for Oracle backup and restore performance.
#
# -----------------------------------------------------------------------------

CMD_STR="
ORACLE_HOME=$ORACLE_HOME
export ORACLE_HOME
ORACLE_SID=$ORACLE_SID
export ORACLE_SID
$RMAN target $TARGET_CONNECT_STR nocatalog send
 \"'NB_ORA_POLICY=$NB_ORA_POLICY, NB_ORA_PC_SCHED=$NB_ORA_PC_SCHED,
 NB_ORA_PC_STREAMS=$NB_ORA_PC_STREAMS, NB_ORA_PC_BTYPE=$NB_ORA_PC_BTYPE'\"
 msglog $RMAN_LOG_FILE append << EOF
RUN {
ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
ALLOCATE CHANNEL ch01 TYPE 'SBT_TAPE';
BACKUP
     PROXY
     SKIP INACCESSIBLE
     TAG hot_db_bk_proxy
     # recommended format
     FORMAT 'bk_%s_%p_%t'
```

```
      DATABASE;
      sql 'alter system archive log current';
RELEASE CHANNEL ch00;
RELEASE CHANNEL ch01;
# backup all archive logs
ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
ALLOCATE CHANNEL ch01 TYPE 'SBT_TAPE';
BACKUP
   FILESPERSET 20
   FORMAT 'al_%s_%p_%t'
   ARCHIVELOG ALL DELETE INPUT;
RELEASE CHANNEL ch00;
RELEASE CHANNEL ch01;
#
# Note: During the process of backing up the database, RMAN also backs up the
# control file.  This version of the control file does not contain the
# information about the current backup because "nocatalog" has been specified.
# Too include the information about the current backup, the control file should
# be backed up as the last step of the RMAN section.  This step would not be
# necessary if we were using a recovery catalog.
#
ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
BACKUP
    # recommended format
    FORMAT 'cntrl_%s_%p_%t'
    CURRENT CONTROLFILE;
RELEASE CHANNEL ch00;
}
EOF
"
# Initiate the command string

if [ "$CUSER" = "root" ]
then
    su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
    RSTAT=$?
else
    /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
    RSTAT=$?
fi

# -----------------------------------------------------------------------------
# Log the completion of this script.
# -----------------------------------------------------------------------------

if [ "$RSTAT" = "0" ]
```

```
then
    LOGMSG="ended successfully"
else
    LOGMSG="ended in error"
fi

echo >> $RMAN_LOG_FILE
echo Script $0 >> $RMAN_LOG_FILE
echo ==== $LOGMSG on `date` ==== >> $RMAN_LOG_FILE
echo >> $RMAN_LOG_FILE

exit $RSTAT
```

Example 5, `hot_tablespace_backup_proxy.sh`

```
# ----------------------------------------------------------------------------
#  hot_tablespace_backup_proxy.sh
# ----------------------------------------------------------------------------
#  This script uses Recovery Manager to take a hot (inconsistent) tablespace
#  backup. A hot backup is inconsistent because portions of the database are
#  being modified and written to the disk while the backup is progressing.
#  You must run your database in ARCHIVELOG mode to make hot backups. It is
#  assumed that this script will be executed by user root. In order for RMAN
#  to work properly we switch user (su -) to the oracle dba account before
#  execution. If this script runs under a user account that has Oracle dba
#  privilege, it will be executed using this user's account.
# ----------------------------------------------------------------------------


# ----------------------------------------------------------------------------
# Determine the user which is executing this script.
# ----------------------------------------------------------------------------

CUSER=`id |cut -d"(" -f2 | cut -d ")" -f1`

# ----------------------------------------------------------------------------
# Put output in <this file name>.out. Change as desired.
# Note: output directory requires write permission.
# ----------------------------------------------------------------------------

RMAN_LOG_FILE=${0}.out

# ----------------------------------------------------------------------------
# You may want to delete the output file so that backup information does
# not accumulate.  If not, delete the following lines.
# ----------------------------------------------------------------------------
```

```
if [ -f "$RMAN_LOG_FILE" ]
then
rm -f "$RMAN_LOG_FILE"
fi

{ # output block

# -------------------------------------------------------------------
# Initialize the log file.
# -------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
chmod 666 $RMAN_LOG_FILE

# ------------------------------------------------------------------------------
# Log the start of this script.
# ------------------------------------------------------------------------------

echo Script $0 >> $RMAN_LOG_FILE
echo ==== started on `date` ==== >> $RMAN_LOG_FILE
echo >> $RMAN_LOG_FILE

# ------------------------------------------------------------------------------
# Replace /db/oracle/product/ora81, below, with the Oracle home path.
# ------------------------------------------------------------------------------

ORACLE_HOME=/db/oracle/product/ora81
export ORACLE_HOME

# ------------------------------------------------------------------------------
# Replace ora81, below, with the Oracle SID of the target database.
# ------------------------------------------------------------------------------

ORACLE_SID=ora81
export ORACLE_SID

# ------------------------------------------------------------------------------
# Replace ora81, below, with the Oracle DBA user id (account).
# ------------------------------------------------------------------------------

ORACLE_USER=ora81

# ------------------------------------------------------------------------------
# Replace productionDB, below, with the target database TNS alias (service)
# name from the tnsnames.ora file.
```

```
                   # ----------------------------------------------------------------------------

                   TARGET_TNS=productionDB
                   export TARGET_TNS

                   # ----------------------------------------------------------------------------
                   # Replace rcvcatDB, below, with the recovery catalog database TNS alias
                   # (service) name from the # tnsnames.ora file.
                   # ----------------------------------------------------------------------------

                   RCVCAT_TNS=rcvcatDB
                   export RCVCAT_TNS

                   # ----------------------------------------------------------------------------
                   # Set the rcvcat connect string.
                   # Replace "rman/rman", below, with the rcvcat connect string.
                   # ----------------------------------------------------------------------------

                   RCVCAT_CONNECT_STR=rman/rman

                   # ----------------------------------------------------------------------------
                   # Set the target connect string.
                   # Replace "sys/manager", below, with the target connect string.
                   # ----------------------------------------------------------------------------

                   TARGET_CONNECT_STR=sys/manager

                   # ----------------------------------------------------------------------------
                   # Set the Oracle Recovery Manager name.
                   # ----------------------------------------------------------------------------

                   RMAN=$ORACLE_HOME/bin/rman

                   # ----------------------------------------------------------------------------
                   # Replace the values of the following variables:
                   #
                   #        NB_ORA_PC_FILESYSTEMS - specifies this number of file systems to be
                   #                        backed up in parallel.  The default is the number
                   #                        of target file systems.  If you do not want to use
                   #                        the default, you will need to pass this variable
                   #                        using the send command.
                   #
                   #        NB_ORA_PC_STREAMS - specifies this number of parallel backup streams to
                   #                        start within each file system.
                   # ----------------------------------------------------------------------------
```

```
#NB_ORA_PC_FILESYSTEMS=1
NB_ORA_PC_STREAMS=3


# -----------------------------------------------------------------------------
# Print out the value of the variables set by this script.
# -----------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
echo    "RMAN: $RMAN" >> $RMAN_LOG_FILE
echo    "ORACLE_USER: $ORACLE_USER" >> $RMAN_LOG_FILE
echo    "ORACLE_HOME: $ORACLE_HOME" >> $RMAN_LOG_FILE
echo    "NB_ORA_PC_STREAMS: $NB_ORA_PC_STREAMS" >> $RMAN_LOG_FILE
echo    "NB_ORA_PC_FILESYSTEMS: $NB_ORA_PC_FILESYSTEMS" >> $RMAN_LOG_FILE

# -----------------------------------------------------------------------------
# Print out the value of the variables passed from the server by bphdb.
# -----------------------------------------------------------------------------

echo >> $RMAN_LOG_FILE
echo    "NB_ORA_SERV: $NB_ORA_SERV" >> $RMAN_LOG_FILE
echo    "NB_ORA_POLICY: $NB_ORA_POLICY" >> $RMAN_LOG_FILE
echo    "NB_ORA_PC_SCHED: $NB_ORA_PC_SCHED" >> $RMAN_LOG_FILE
echo    "NB_ORA_PC_BTYPE: $NB_ORA_PC_BTYPE" >> $RMAN_LOG_FILE
echo >> $RMAN_LOG_FILE

# -----------------------------------------------------------------------------
# This script assumes that the database is properly opened. If desired,
# this would be the place to verify that.
# -----------------------------------------------------------------------------


# -----------------------------------------------------------------------------
# Call Recovery Manager to initiate the backup. This example uses a
# Recovery Catalog. If you do not want to use a recovery catalog, add the
# command 'nocatalog' to the rman command line below and remove the
# 'rcvcat <userid>/<passwd>@<tns alias>' statement.
#
# Use the send command to send NBU policy, schedule name and the number of
# the backup streams to be used in the database proxy backup.
# -----------------------------------------------------------------------------

CMD_STR="
ORACLE_HOME=$ORACLE_HOME
export ORACLE_HOME
ORACLE_SID=$ORACLE_SID
export ORACLE_SID
$RMAN target $TARGET_CONNECT_STR@TARGET_TNS rcvcat
  $RCVCAT_CONNECT_STR@RCVCAT_TNS send \"'NB_ORA_POLICY=$NB_ORA_POLICY,
```

```
  NB_ORA_PC_SCHED=$NB_ORA_PC_SCHED, NB_ORA_PC_STREAMS=$NB_ORA_PC_STREAMS,
  NB_ORA_PC_BTYPE=$NB_ORA_PC_BTYPE'\" msglog $RMAN_LOG_FILE append << EOF
RUN {
ALLOCATE CHANNEL ch00 TYPE 'SBT_TAPE';
BACKUP
    PROXY
    SKIP INACCESSIBLE
    TAG hot_tbs_bk_proxy
    # recommended format
    FORMAT 'df_%s_%p_%t'
    TABLESPACE SYSTEM;
    sql 'alter system archive log current';
RELEASE CHANNEL ch00;
}
EOF
"
# Initiate the command string

if [ "$CUSER" = "root" ]
then
    su - $ORACLE_USER -c "$CMD_STR" >> $RMAN_LOG_FILE
    RSTAT=$?
else
    /usr/bin/sh -c "$CMD_STR" >> $RMAN_LOG_FILE
    RSTAT=$?
fi

# ----------------------------------------------------------------------------
# Log the completion of this script.
# ----------------------------------------------------------------------------

if [ "$RSTAT" = "0" ]
then
    LOGMSG="ended successfully"
else
    LOGMSG="ended in error"
fi

echo >> $RMAN_LOG_FILE
echo Script $0 >> $RMAN_LOG_FILE
echo ==== $LOGMSG on `date` ==== >> $RMAN_LOG_FILE
echo >> $RMAN_LOG_FILE

exit $RSTAT
```

Index