

SnapManager® 2.0 for Oracle®

Installation and Administration Guide

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089 USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: <http://www.netapp.com>

Part number 215_02138_A0
December 2006

Copyright and trademark information

Copyright information

Copyright © 1994–2006 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted Network Appliance material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETWORK APPLIANCE “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL NETWORK APPLIANCE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp, the Network Appliance logo, the bolt design, NetApp—the Network Appliance Company, DataFabric, Data ONTAP, Decru, FAServer, FilerView, MultiStore, NearStore, NetCache, SecureShare, SnapDrive, SnapLock, SnapManager, SnapMirror, SnapMover, SnapRestore, SnapValidator, SnapVault, Spinnaker Networks, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, SyncMirror, VFM, and WAFL are registered trademarks of Network Appliance, Inc. in the U.S.A. and/or other countries. gFiler, Network Appliance, SnapCopy, Snapshot, and The evolution of storage are trademarks of Network Appliance, Inc. in the U.S.A. and/or other countries and registered trademarks in some other countries. ApplianceWatch, BareMetal, Camera-to-Viewer, ComplianceClock, ComplianceJournal, ContentDirector, ContentFabric, Decru DataFort, EdgeFiler, FlexClone, FlexShare, FlexVol, FPolicy, HyperSAN, InfoFabric, LockVault, Manage ONTAP, NOW, NOW NetApp on the Web, ONTAPI, OpenKey, RAID-DP, RoboCache, RoboFiler, SecureAdmin, Serving Data by Design, SharedStorage, Simplicore, Simulate ONTAP, Smart SAN, SnapCache, SnapDirector, SnapFilter, SnapMigrator, SnapSuite, SohoFiler, SpinMirror, SpinRestore, SpinShot, SpinStor, StoreVault, vFiler, Virtual File Manager, VPolicy, and Web Filer are trademarks of Network Appliance, Inc. in the United States and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of Network Appliance, Inc. in the U.S.A.

Apple is a registered trademark and QuickTime is a trademark of Apple Computer, Inc. in the United States and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the United States and/or other countries. RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the United States and/or other countries.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Network Appliance is a licensee of the CompactFlash and CF Logo trademarks.

Network Appliance NetCache is certified RealSystem compatible.

Table of Contents

	Preface	ix
Chapter 1	SnapManager for Oracle Overview	1
	What SnapManager for Oracle Can Do for You	2
	How SnapManager for Oracle Works	4
	SnapManager Glossary	7
Chapter 2	SnapManager Concepts.	9
	Basic SnapManager Concepts	10
	Working with repositories	12
	Approaches to backing up, restoring, and recovering data	14
	How SnapManager handles backups	19
	SnapManager's policy for retaining backups.	23
	SnapManager operation states	25
Chapter 3	SnapManager Deployment Considerations	27
	SnapManager for Oracle requirements	28
	Prerequisites, considerations and limitations	29
Chapter 4	SnapManager Interfaces	33
	Interface choices	34
	Using the SnapManager for Oracle GUI	35
Chapter 5	Installing or Upgrading SnapManager	41
	Installing and upgrading SnapManager tasks.	42
	Downloading the SnapManager software package.	44
	Installing or upgrading the SnapManager software packages	46
	Verifying, starting and stopping the server.	49

Chapter 6	Updating and Creating Repositories	51
	Updating and creating repositories	52
Chapter 7	Quick Start Example Procedure	55
	Before you begin	56
	Example procedure	58
Chapter 8	Creating Profiles.	61
	Verifying the environment	62
	Creating a profile	63
	Verifying the profile	66
Chapter 9	Backing Up and Restoring Databases	67
	Creating database backups	68
	Verifying a backup	71
	Restoring and recovering a database backup	73
Chapter 10	Cloning Databases Using SnapManager.	77
	General SnapManager database cloning information	78
	Creating a clone specification	80
	Cloning a database from a backup	88
	Cloning a database in its current state	90
	Cloning with RAC databases	91
Chapter 11	Managing SnapManager Operations	93
	Aborting an operation.	94
	Deleting repositories	97
	Changing an existing profile	98
	Deleting a profile	99
	Displaying information about backups	100

	Deleting backups	101
	Mounting and unmounting backups	103
	Checking the version of SnapManager	106
	Uninstalling SnapManager software	107
Chapter 12	Advanced Topics	109
	Performing a block level restore using RMAN.	110
Chapter 13	Troubleshooting	115
	Questions to consider.	116
	Gathering troubleshooting information.	118
	General troubleshooting tips for using the GUI	121
	General troubleshooting tips for working with clones	122
	General troubleshooting tips for SnapManager	124
	Common error conditions.	129
	Where to go for more information	131
Chapter 14	SnapManager Command Reference	133
	Collecting information needed by SnapManager Commands	134
	Conventions for entering names and command options	135
	Common options for handling messages shown on the console	136
	The smo backup create command	137
	The smo backup delete command	142
	The smo backup list command	144
	The smo backup mount command	146
	The smo backup restore and recovery command.	148
	The smo backup unmount command	151
	The smo backup verify command	153
	The smo clone create command	155
	The smo clone template command	157

The smo cmdfile command	.158
The smo help command	.160
The smo operation abort command.	.161
The smo operation dump command	.163
The smo profile create command.	.165
The smo profile delete command.	.170
The smo profile dump command	.172
The smo profile list command	.173
The smo profile show command	.174
The smo profile update command	.176
The smo profile verify command.	.181
The smo repository create command	.182
The smo repository delete command	.184
The smo repository update command	.186
The smo system dump command.	.188
The smo system verify command.	.189
The smo version command	.190
Index	.191

Preface

- About this guide

This document describes how to install, configure, and use SnapManager® 2.0 for Oracle® software to back up, restore, and clone databases.

It does not cover basic system or network administration topics, such as IP addressing, routing, and network topology. It also does not cover topics that are handled in the Network Appliance™ FCP Host Attach Kit, iSCSI Support Kit, Open Systems SnapVault®, or SnapDrive® for UNIX® documentation.
- Audience

This guide is for database administrators who are familiar with Oracle and storage administrators who possess a working knowledge of NetApp® storage systems.
- Command conventions

SnapManager for Oracle (SnapManager) lets you enter commands from the Command Line Interface (CLI) on your UNIX host where the target database is running. You can also execute SnapManager from a Graphical User Interface (GUI) running on a Windows host.
- Formatting conventions

The following table lists different character formats used in this guide to set off special information.

Formatting convention	Type of information
<i>Italic type</i>	◆ Words or characters that require special attention. ◆ Placeholders for information you must supply. For example, if the guide requires you to enter the <code>fcetest adaptername</code> command, enter the characters “fcetest” followed by the actual name of the adapter. ◆ Book titles in cross-references.
Monospaced font	◆ Command and daemon names. ◆ Information displayed on the system console or other computer monitors. ◆ The contents of files.

Formatting convention	Type of information
Bold monospaced font	Words or characters you type. What you type is always shown in lowercase letters, unless your program is case-sensitive and uppercase letters are necessary for it to work properly.

Keyboard conventions

This guide uses capitalization and some abbreviations to refer to the keys on the keyboard. The keys on your keyboard might not be labeled exactly as they are in this guide.

What is in this guide...	What it is used to mean...
hyphen (-)	Separates individual keys. For example, Ctrl-D means holding down the Ctrl key while pressing the D key.
<i>Enter</i>	Refers to the key that generates a carriage return; the key is named Return on some keyboards.
<i>type</i>	Pressing one or more keys on the keyboard.
<i>enter</i>	Pressing one or more keys and then pressing the Enter key.

Special messages

This guide contains special messages that are described as follows:

Attention

An attention notice contains instructions that you must follow to avoid a system crash, loss of data, or damage to the equipment.

Note

A note contains important information that helps you install or operate the system efficiently.

About this chapter

The SnapManager® for Oracle® features benefit both database administrators (DBAs) and storage administrators.

This topics in this chapter are:

- ◆ [“What SnapManager for Oracle Can Do for You”](#) on page 2
- ◆ [“How SnapManager for Oracle Works”](#) on page 4
- ◆ [“SnapManager Glossary”](#) on page 7

What SnapManager for Oracle Can Do for You

SnapManager from a DBA's perspective

SnapManager for Oracle makes backups easier for Database Administrators (DBA). SnapManager for Oracle:

- ◆ Creates a backup quickly and in a space-efficient way, which lets you perform more backups.
- ◆ Organizes information into groupings, such as profiles, to make creating backups and restoring and recovering data quick and easy.
- ◆ Automatically maps the database files to the storage. You no longer need to know the underlying storage system.
- ◆ Integrates with existing Oracle tools, such as Recovery Manager (RMAN) and Automatic Storage Management (ASM).
- ◆ Creates snapshots of logs.
- ◆ Quickly creates a clone from a backup.
- ◆ Reduces the mean time to recover a database by using SnapRestore.

SnapManager from a storage administrator's perspective

SnapManager for Oracle makes managing the storage required for backups easier for a storage administrator. SnapManager for Oracle:

- ◆ Handles different protocols (FCP, iSCSI, NFS).
- ◆ Gives you options to optimize backups based on the type of backup (complete or partial) that works best in your environment.
- ◆ Makes backups quick and space efficient, so you can do them more frequently, if necessary.
- ◆ Creates quick and space efficient clones.
- ◆ Works with host volume managers, such as Automatic Storage Manager (ASM), VERITAS, or Logical Volume Manager (LVM).

SnapManager is a stand-alone product that integrates features from other NetApp products to enable it to produce fast backups that require only a small amount of space. It also works with Oracle products to enable DBAs to continue using their current tool sets.

The products SnapManager works with include:

- ◆ SnapDrive for UNIX. It uses this product to create snapshots of the storage. Using snapshots ensures that the backups are space efficient and faster to create than disk-to-disk backups.

- ◆ FlexClone feature, which is part of Data ONTAP. SnapManager for Oracle uses FlexClones to create fast, space-efficient clones of backups.

How SnapManager for Oracle Works

SnapManager simplifies the backup of data in Oracle databases. Once you have set up a profile containing the credentials for your database instance, you can create a backup using a single CLI command on your UNIX host.

SnapManager also provides a GUI for installation on a Windows host. The GUI allows you to manage SnapManager operations from a remote location, and uses screens and wizards to guide you through the operations. The same operations can be performed through the GUI as the CLI.

SnapManager for Oracle Tasks

SnapManager works with storage systems running Data ONTAP to enable you to perform the following tasks:

- ◆ Use snapshots to create a backup of data on a storage system running Data ONTAP.

Backups using snapshots take much less time to create and require significantly less space than full disk-to-disk backups. This is because the SnapManager backup is a virtual copy of the database and is stored on the same physical medium as the database.

Create backups of a single database instance, one or more tablespaces, or one or more data files (such as data, log, and control files). This data can be spread across different storage systems, storage system volumes, or LUNs. You can also use SnapManager to back up a database when there are multiple databases in the same volume or LUN.

SnapManager also works with databases configured with Oracle's Automatic Storage Management (ASM) feature and Oracle's Recovery Manager (RMAN).

Note

To make this document easier to read, it uses the term database to refer to SnapManager operations that affect the database instance or portions of the database instance. It only uses tablespaces and files in cases where it is important to distinguish between database instances, tablespaces, and files.

- ◆ Restore and recover a full database or a portion of the database. SnapManager restores the full database instance, specific tablespaces, specific files, or control files. Once the information is restored, use SnapManager's recover operation to put the database in a usable state. You also have the choice of using SnapManager to restore the database and

another tool to recover the information; you are not restricted to using SnapManager for both operations.

Specify the level to which SnapManager restores and recovers information when performing the restoration. For example, you can recover based on a point in time. This can be either a date and time or an Oracle System Change Number (SCN).

- ◆ Track details and produce reports.

SnapManager reduces the level of detail you must track. When you specify the database to back up, SnapManager automatically identifies the database files for the backup. You do not need to worry about the underlying database in terms of host file systems, host volumes, and so on.

In addition, SnapManager displays information about the backup, the profile, and the repository.

- ◆ Verify the validity of the backup.

Use SnapManager to confirm the integrity of the backup using standard Oracle backup verification operations. Perform the verification as part of the backup operation or at a later time.

- ◆ Clone a backup of a database.

SnapManager quickly creates a writable, space-efficient clone of a database instance it has backed up. The clone is separate from the actual backup, so you can modify it without changing the backup source. SnapManager uses the FlexClone feature, which is part of Data ONTAP, to create the clone. FlexClone enables SnapManager to use snapshots of the database instance to avoid having to create an entire physical, disk-to-disk copy. The snapshots require less time to create and take up significantly less space than a physical copy.

ASM, RAC, and RMAN integration

To make tasks go smoothly for DBAs familiar with Oracle, SnapManager also provides an integration point for such Oracle capabilities as ASM, Real Application Clusters (RAC), and RMAN.

Invoke SnapManager from a RAC database node to create the backup, restore and recover the database, and clone the database.

SnapManager can catalog the backups it makes with RMAN. If using RMAN, the DBA can make use of SnapManager's backups and preserve the value of all RMAN functions, such as block level restore.

SnapManager lets RMAN use the snapshots when it performs recovery or restore. For example, you can use RMAN to restore a table within a tablespace as well as perform full database and tablespace restores and recoveries from snapshots taken by SnapManager.

Overview of repositories and profiles

SnapManager organizes information into profiles and repositories. The profiles hold the information about the database, including its credentials, backups, and clones. As a result, you only need to supply a profile name when performing SnapManager operations; after profile (database) authentication you do not need to repeat the credentials required to access the information.

Repositories hold the profiles and data about operations performed on the profiles. The repository must be located in a different database from the one you are backing up. The database holding the repository can be on the same storage system as the database you are backing up or it can be on a different system. A single repository can hold information on multiple profiles; however, each database is normally associated with only one profile. You can have multiple repositories, where each repository holds multiple profiles.

When you restore a database or recover a portion of it, SnapManager queries the repository to determine what was backed up.

For more information on profiles and repositories, see “[SnapManager Concepts](#)” on page 9.

SnapManager Glossary

Backup: A logical snapshot of a database, one or more tablespaces, or one or more files that you can use to restore this information.

Full backup: SnapManager makes a copy of the entire database instance.

GUID: A Globally Unique Identifier for each operation that SnapManager automatically generates for each operation.

Offline: The database is not available to users (i.e., it is SHUTDOWN, STARTED, or MOUNTED).

Online: The database is available to users (i.e., OPEN).

Partial backup: SnapManager makes a copy of either the tablespaces or files you specify.

Profile: The metadata that describes the environment for SnapManager operations. Each profile is associated with a database. SnapManager uses profiles to hold the database and operating system credentials and information on operations, including database backups, clones, and restorations.

Recover: An operation that enables you to use a restored database instance. The recover operation is part of the SnapManager restore command. Perform a SnapManager restore operation and use a different tool to recover the database, or perform a SnapManager restore/recover operation. SnapManager recovers the database to a known state from a given backup when the recover option is included.

Repository: A container used by SnapManager to persistently store metadata about backup, restore, and clone operations. This includes the date and time of the backup, the data files included in the backup, and the name of the NetApp storage system. This information is organized based on the profile of the database instance. Each repository can hold multiple profiles.

Restore: An operation that replaces the current data (data files, control files, and so on) with the backed up data.

Snapshot: A Snapshot copy is a point-in-time, read-only image of the storage system volume.

About this chapter

This chapter provides information about SnapManager concepts.

This chapter includes the following topics:

- ◆ “[Basic SnapManager Concepts](#)” on page 10
- ◆ “[Working with repositories](#)” on page 12
- ◆ “[Approaches to backing up, restoring, and recovering data](#)” on page 14
- ◆ “[How SnapManager handles backups](#)” on page 19
- ◆ “[SnapManager’s policy for retaining backups](#)” on page 23
- ◆ “[SnapManager operation states](#)” on page 25

Basic SnapManager Concepts

How SnapManager uses repositories and profiles

SnapManager organizes database and backup information into repositories and profiles. These are the groupings to use when backing up and restoring databases.

The SnapManager GUI uses the repositories and profiles as the navigation tool to help locate the tasks to perform. For example, to create a backup, use the Navigation panel of the GUI to select the repository. From the repository, select the profile associated with that database. Then perform the backup operation.

Understanding repositories

The repository is the storage container for the metadata that SnapManager uses when working with backups. It records such information as when a backup took place, which files were backed up, and whether a clone was created from the backup.

Create a repository schema against an Oracle database. This can be an existing Oracle database or one created for this purpose.

Once SnapManager is running, create additional repositories, backup or delete repositories.

Each repository can hold multiple profiles. The repository contains information about the profile and its backups, so it cannot be located in the same database as the one associated with the profile. The repository can exist on either the same storage system as the profiled database or a different storage system.

Understanding profiles

The profile contains the information required for creating the backup and for performing other SnapManager operations, as well as information on the operations that have been performed. This information includes the database credentials and how many backups to retain. The credentials stored include the username/password pairs for accessing the repository, the database, and the required connection information for using either RMAN or ASM.

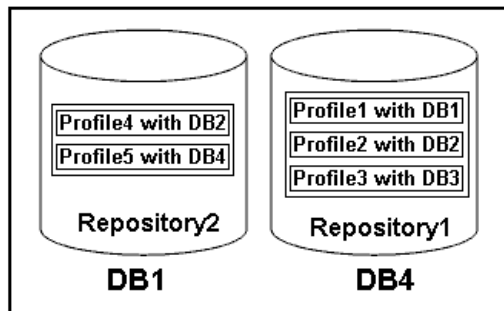
SnapManager requires that you create a profile for each database. A profile can contain only one database. That same database can be defined in more than one profile.

SnapManager places a lock on the database during an operation to prevent two incompatible operations from being performed simultaneously. You can abort some operations. Refer to [“Aborting an operation”](#) on page 94.

The profile also serves as the container for operations such as backups, clones, etc.. You cannot access a backup created using one profile from a different profile, even if both profiles are associated with the same database.

Once a profile is created, you can perform SnapManager operations. Specify the profile name with the SnapManager command; you do not need to re-enter the database credentials each time.

The figure below illustrates how repositories can hold multiple profiles, but each profile can define only one database. In this example, Repository2 is on the database DB1 and Repository1 is on the database DB4.



If using RAC, specify one of the RAC database instances from the cluster as the node SnapManager operates on. This instance must be running on the host specified.

Working with repositories

Using multiple repositories to make backups of repositories

Use one repository to handle multiple profiles, although it's a good practice to have at least two repositories that reside in separate databases. That way, you can create backups of each repository by using a profile that is stored in the opposite repository.

For example, if Repository1 is in Database1, a backup of it is made each time you create a backup of Database1. Conversely, a backup is created of Repository2 each time you create a backup of Database2. Backing up the repositories this way protects against a single database failure.

Example of making a backup of the repository

To create a backup of a repository, first determine which profile is associated with the database that contains the repository. Now make a backup of that database.

For example, Repository1 is located in the database DB4d, and Repository2 is located in the database DB1a.

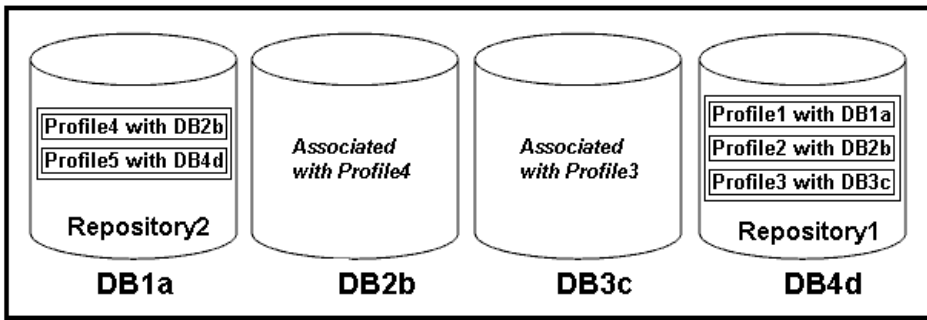
Repository1 has three profiles:

- ◆ Profile1, which is associated with database DB1a.
- ◆ Profile2, which is associated with DB2b.
- ◆ Profile3, which is associated with DB3c.

Repository2 has two profiles:

- ◆ Profile4, which is associated with database DB2b.
- ◆ Profile5, which is associated with DB4d.

The repository and profiles appear on the databases as shown in the following diagram.



The repositories are only on two of the databases. To create a backup of each repository, use a profile that contains the repository. For example, because Repository1 is in database DB4d, create a backup of Repository1 by backing up Profile5, which is associated with DB4d.

Repository2 is in database DB1a, so create a backup of it by backing up Profile1, which is associated with that database.

Both repositories contain a profile associated with Database2, so neither repository could be located in Database2. SnapManager does not allow the target database to be the same as the repository database.

Approaches to backing up, restoring, and recovering data

Types of backup, restore, and recovery operations

SnapManager provides several options for backing up, restoring, and recovering data. Perform these operations using the entire backed up database instance or portions of it. SnapManager offers the following choices:

- ◆ Backup operations. Each backup consists of one of the following:
 - ❖ The entire database instance
 - ❖ A group of tablespaces
 - ❖ A group of data files

Note

SnapManager includes the control files and archived log files with each backup. To see which tablespaces and data files are included in a backup, use the `smo backup list` command.

- ◆ Restore operations. Restore the entire backup or a portion of it. If you perform a selective restore operation, specify one of the following:
 - ❖ A group of tablespaces
 - ❖ A group of data files
 - ❖ The control files

Note

If you specify a complete restore operation, SnapManager replaces only the data files. The control files and log files are not part of a complete restore operation.

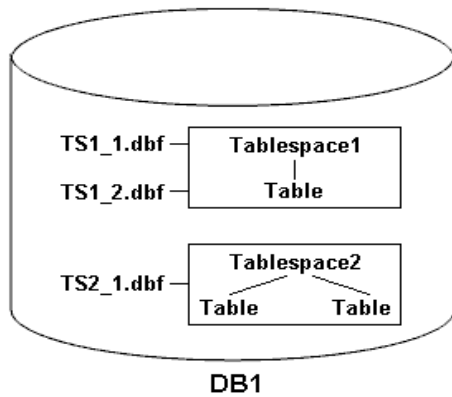
- ◆ Recovery operations. Use SnapManager to recover the restored data, by specifying that it recover the data based on either a point in time or all available logs (i.e., the last transaction committed to the database). The point in time can be an Oracle SCN or a date and time (`yyyy-mm-dd:HH:MM:SS`). SnapManager uses the 24-hour clock.

The following table summarizes the options available with the SnapManager backup, restore, and recover operations. Use SnapManager or another tool to perform the recovery.

Backup	Restore	Recover using SnapManager
Entire database instance Note _____The database must be in archive log mode to create a backup while the database is running (i.e., online)._____	Entire backup	◆ To a specific date ◆ To an SCN ◆ To the time of the backup ◆ To the last transaction made on the database instance
Entire database instance	One of the following: ◆ Tablespaces ◆ Data files ◆ Control files	◆ To the last transaction made to the database instance
One of the following: ◆ Tablespaces ◆ Data files	Entire backup (this is either a set of tablespaces, a set of data files, or the control files)	◆ To the last transaction made to the database instance
One of the following: ◆ Tablespaces ◆ Data files	One of the following: ◆ Tablespaces ◆ Data files ◆ Control files	◆ To the last transaction made to the database instance

Examples of these operations

The database has two tablespaces. Tablespace1 has one table and two database files associated with it. Tablespace2 has two tables and one database file associated with it.



There are several options for creating backups of this data and then restoring and recovering the data. The following table describes some backup and restore/recover scenarios.

Backup	Restore	Recover
Full backup: SnapManager makes a backup of DB1, including the data files, archive logs, and control files.	Complete restore: SnapManager restores all data files in the backup.	◆ SnapManager recovers the data to the point in time. Specify: ❖ SCN. Enter an SCN, such as 384641. ❖ Date/time. Enter a date, such as, 2005-11-25:19:06:22. ❖ To the last transaction made to the database. ❖ Using a manual recovery in Oracle instead of using SnapManager's recover operation.

Backup	Restore	Recover
Full backup: SnapManager makes a backup of everything in DB1, including the data files, archive logs, and control files.	Selective restore: Specify which information SnapManager should restore. ◆ Tablespaces. Specify both Tablespace1 and Tablespace2 or only one of them. ◆ Data files. Specify all three database files (TS1_1.dbf, TS1_2.dbf, and TS2_1.dbf), two of these files, or one of these files. ◆ Control files. SnapManager restores only the control files.	◆ SnapManager recovers the data to the last transaction made to the database. ◆ Using a manual recovery in Oracle instead of using SnapManager's recover operation.
Partial backup: Specify the data that SnapManager puts in the backup. Choose one of the following options: ◆ Tablespaces. Specify both Tablespace1 and Tablespace2 or only one of them. ◆ Data files. Specify all three database files (TS1_1.dbf, TS1_2.dbf, and TS2_1.dbf), two of these files, or one of these files. Regardless of which option you choose, the backup includes the archive logs and control files.	Complete restore: SnapManager restores all data files in the backup.	◆ To the last transaction made to the database instance. ◆ Using a manual recovery in Oracle instead of using SnapManager's recover operation.

Backup	Restore	Recover
Partial backup: Specify the data SnapManager puts in the backup. Choose one of the following options: ◆ Tablespaces. Specify both Tablespace1 and Tablespace2 or only one of them. ◆ Data files. Specify all three database files (TS1_1.dbf, TS1_2.dbf, and TS2_1.dbf), two of these files, or one of these files. Regardless of which option you choose, the backup includes the archive logs and control files.	Selective restore: SnapManager restores a subset of the partial backup. If the backup contains Tablespace1 and Tablespace2, SnapManager would restore only the tablespace specified. If it contains all three database files (TS1_1.dbf, TS1_2.dbf, and TS2_1.dbf), SnapManager would restore only the data files specified. If you enter the option <code>-controlfiles</code>, SnapManager only restores the control files.	◆ To the last transaction made to the database instance. ◆ Using a manual recovery in Oracle instead of using SnapManager's recover operation.

How SnapManager handles backups

The effect of choosing a full or partial online backup

SnapManager uses different algorithms for creating backups, based on whether you perform a full backup, or a partial backup containing only certain data files or tablespaces.

Based on the algorithm SnapManager uses for the backup, it optimizes either the number of snapshots or amount of time the database is in online backup mode:

- ◆ Performing a complete online backup minimizes the number of snapshots. SnapManager takes one snapshot for each volume that the database uses plus one snapshot for each volume that the log files occupy.

For online backups, this means each tablespace is in backup mode for the entire time that the backup operation takes, which may be undesirable. The advantage is that this algorithm ensures SnapManager takes the minimum number of snapshots.

- ◆ Performing a partial online backup minimizes the amount of time each tablespace is in backup mode. SnapManager groups the snapshots it takes by tablespace. Each tablespace is in backup mode only long enough to take the snapshots, so this method of grouping the snapshots can minimize the physical block writes in the log files during an online backup. However, since the backup could require taking snapshots of multiple tablespaces in the same volume, this method could cause SnapManager to take multiple snapshots of a single volume during the backup operation. As a result, more snapshots may be created.

Note

Data ONTAP limits the maximum number of snapshots to 255 per volume. You are only likely to reach this maximum if SnapManager is set up to retain a large number of backups where each backup consists of numerous snapshots.

The effect of backup choices on snapshots

The following example compares the differences in how the algorithms create snapshots using a database consisting of two volumes. Each volume has two tablespaces and each tablespace has two database files.

For a full backup, SnapManager takes a snapshot of each volume, resulting in a total of two snapshots.

For a partial backup of the same number of database objects, SnapManager takes a snapshot of each tablespace file, resulting in four snapshots.

Note

Both backups would also take snapshots of the log files.

The following table shows how the two types of backups produce a different number of snapshots.

		Partial Backup		
		↓ Tablespace TS1	↓ Tablespace TS2	
Full Backup →	/vol/vola	ts1_1.dbf	ts2_1.dbf	→ 1 snap
	/vol/volb	ts1_2.dbf	ts2_2.dbf	→ 1 snap
		↓ 2 snaps	↓ 2 snaps	

Select the type of backup based on what is important in your work environment. Consider the acceptable length of time for the database to be in backup mode versus the number of snapshots available on the volume for use.

For more details on these backup algorithms, see “[Algorithm for performing a full backup](#)” on page 20 and “[Algorithm for performing a partial backup](#)” on page 21.

Algorithm for performing a full backup

When performing a full backup, SnapManager places the entire database in online backup mode. An individual tablespace (for example, /vol/vola/ts1_1.dbf) is in online backup mode longer than if certain tablespaces or data files were specified.

Since databases do more work in online backup mode, this places a greater load on the host. When a database goes into backup mode, it writes entire blocks to the logs, instead of simply writing the delta of the change to a log file. This equates to additional I/O load on the host, however, SnapManager takes fewer snapshots.

When you specify a full backup, SnapManager:

1. Places the entire database in online backup mode.

2. Takes snapshots of all the volumes containing database files.
3. Takes the database out of online backup mode.
4. Forces a log switch and then archives the log file.
5. Generates backup control files
6. Takes a snapshot of the log files and the backup control files.

SnapManager takes snapshots at the volume level instead of the tablespace level, which usually reduces the number of snapshots it must take.

Algorithm for performing a partial backup

Selecting a partial backup causes SnapManager to process the backup one tablespace at a time, even if you specified that it back up data files. SnapManager uses tablespaces because the tablespace level is the lowest level that Oracle allows into backup mode. Oracle does not permit a file to be put in backup mode.

With a partial backup, each tablespace is in backup mode the shortest amount of time. However, this process requires more snapshots than performing a full backup, increasing the risk of reaching the limit of the number of snapshots allowed sooner.

SnapManager takes snapshots of specific tablespaces or data files. The partial backup algorithm is a loop that SnapManager repeats until it has taken a snapshot of each specified tablespace or data file. During a partial backup, SnapManager:

1. While there are tablespaces or data files to be backed up:
 - ❖ Puts the tablespace containing the data files into backup mode.
 - ❖ Takes a snapshot of all the volumes used by the tablespace.
 - ❖ Takes the tablespace out of backup mode.
 - ❖ It continues this process until it has taken a snapshot of all the tablespaces or files.
2. Forces a log switch and then archives the log file.
3. Generates backup control files
4. Takes a snapshot of the log files and the backup control files.

When performing an online backup, the database is always available to users; however, the database must perform more work and the host must perform more physical I/O. In addition, because it is taking snapshots of each tablespace specified or each tablespace containing a specified data file instead of the entire volume, SnapManager takes more snapshots.

Reducing the number of snapshots on your system

Removing backups when they are no longer needed helps you keep an adequate pool of available snapshots while reducing the risk of hitting the limit for the maximum number of snapshots. Set up SnapManager to automatically remove successful backups once it reaches a certain number of them using SnapManager's retain policy. See "[SnapManager's policy for retaining backups](#)" on page 23 for more details.

SnapManager’s policy for retaining backups

How SnapManager determines which backups to retain

- SnapManager lets you specify how many successful backups it should retain when the profile for that database is set up. This number is affected by the following rules:
- ◆ It applies only to successful backups. SnapManager does not count unsuccessful backups when calculating which backups to keep.
SnapManager keeps information on both successful and unsuccessful backups. The unsuccessful backups contain metadata about the backup and take up only minimal space in the repository.
 - ◆ It counts backups that were used to create clones, but it does not delete them. SnapManager keeps all backups used to create a clone.
 - ◆ SnapManager always keeps the backup it just took.

Note _____
To delete an unsuccessful backup or a cloned backup, use SnapManager’s backup delete operation.

Set the retain option to have SnapManager keep a number of backups. Once it has a successful backup that exceeds this number, it removes the oldest successful backup that was not used to create a clone.

Example of how SnapManager retains backups

SnapManager uses a sliding window approach to retaining backups. For example, the retain option in the profile is set to 3. Make a backup using the profile each week and you have the following backups:

3/3	3/10	3/17	3/24	3/31
Successful	Cloned	Failed	Successful	Cloned
Delete	Keep	Keep	Keep	Keep

In this case, SnapManager would delete the 3/3 backup. It would keep the other four because one of them was a failed backup and does not count.

On April 7, take another backup that is successful. The current list of backups now looks like this:

3/10	3/17	3/24	3/31	4/7
Cloned	Failed	Successful	Cloned	Successful
Keep	Keep	Delete	Keep	Keep

The number of backups listed is now five. SnapManager does not delete a successful backup used as a clone, so it deletes the 3/24 success backup and keeps the 3/10 backup.

On April 14, take another backup.

3/10	3/17	3/31	4/7	4/14
Cloned	Failed	Cloned	Successful	Successful
Keep	Keep	Keep	Delete	Keep

Because this backup is successful, SnapManager deletes the successful backup taken 4/7. This way it is retaining the maximum of three successful backups (3/10, 3/31, and 4/14).

Take two more backups in April that are both successful.

3/10	3/17	3/31	4/14	4/21	4/28
Cloned	Failed	Cloned	Successful	Successful	Successful
Keep	Keep	Keep	Delete	Delete	Keep

SnapManager deletes the oldest successful backups that were not used to create a clone. This keeps the total number of successful backups to three. The number backups is now four:

- ◆ 3/10, which was used to make a clone
- ◆ 3/17, which failed
- ◆ 3/31, which was used to make a clone
- ◆ 4/28, which was successful

SnapManager operation states

Four possible SnapManager operation states

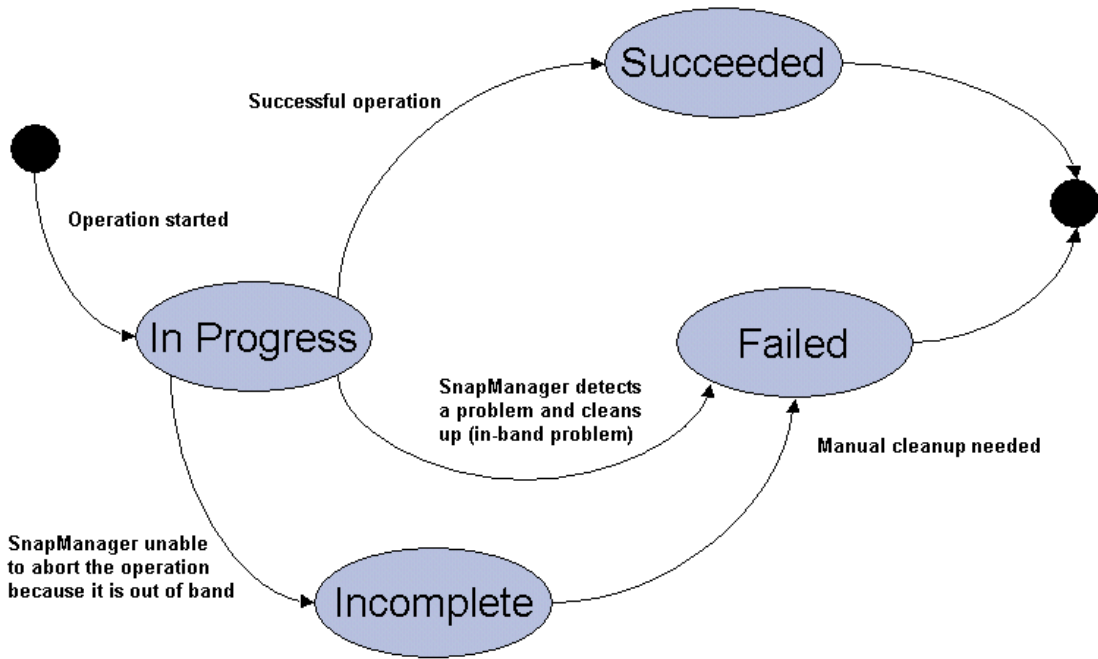
When executing a SnapManager command, there are four possible operation states that it can enter. They are:

- ◆ In progress. An operation, such as a backup operation, is running.
- ◆ Succeeded. The operation completed successfully.
- ◆ Incomplete. An event occurred that caused the operation to fail. When the state is incomplete, it means an unrecoverable (i.e., out-of-band) event occurred. In this case, SnapManager cannot perform its standard abort processing. You must manually abort the operation by using the `smo operation abort` command.

```
# smo operation abort -id id_number
```

- ◆ Failed. The operation failed. If it is a recoverable (i.e., in-band) operation, SnapManager automatically executes its abort process and cleans up the operation.

The following illustration shows the operation states for a command:



Recoverable events

In general, a recoverable (or in-band) event occurs when one of the following happens:

- ◆ SnapManager detects an environment issue. This can include problems such as:
 - ❖ The database is not stored on a storage system running Data ONTAP.
 - ❖ You are using an ASM database, but the ASM instance is not running.
 - ❖ SnapDrive for UNIX is not installed or cannot access the storage system.
- ◆ SnapManager fails to create a snapshot or provision storage. This might be because:
 - ❖ The volume is out of space.
 - ❖ The maximum number of snapshots has been reached.
- ◆ An unanticipated exception occurs.

Recoverable events

When a recoverable event happens, SnapManager performs an abort process. It attempts to return the host, database, and storage system to its starting state. If the abort process fails, SnapManager then treats the incident as an unrecoverable event.

Unrecoverable events

An unrecoverable (or out-of-band) event occurs when any of the following happens:

- ◆ A system issue occurs, such as when a host fails.
- ◆ The SnapManager process is killed.
- ◆ An in-band abort operation fails. This could happen because of one of the following:
 - ❖ The storage system fails.
 - ❖ The LUN or storage volume is offline.
 - ❖ The network fails

Out-of-band events

When an out-of-band event occurs, SnapManager cannot execute the abort process, so it kills the process. This leaves SnapManager in an incomplete operation state. To clear this state, first fix the underlying issue manually (i.e., on the storage volume) and then enter the `smo operation abort` command. Enter the process number as the value for the `id` argument. Use the format:

```
# smo operation abort -id id_number
```

About this chapter

This chapter explains some areas to consider before using SnapManager for Oracle.

This chapter includes information about:

- ◆ “[SnapManager for Oracle requirements](#)” on page 28
- ◆ “[Prerequisites](#)” on page 29
- ◆ “[Topologies](#)” on page 30
- ◆ “[Oracle limitations](#)” on page 31

SnapManager for Oracle requirements

The following are requirements and considerations for using SnapManager in your environment:

- ◆ SnapDrive for UNIX is a requirement. SnapManager uses the features in this product, so it must be installed before you can run SnapManager. SnapManager handles all the interactions with SnapDrive for UNIX. SnapDrive for UNIX must be configured correctly for your storage system and protocol choices.
- ◆ A FlexClone™ license is required to take full advantage of SnapManager with NFS databases. SnapManager uses FlexClone to:
 - ❖ Mount backups of NFS databases.
 - ❖ Verify backups of NFS databases.
 - ❖ Clone NFS databases.
 - ❖ Register backups of NFS databases with RMAN (if using RMAN).FlexClone is part of the Data ONTAP® version supported by SnapManager. As with SnapDrive for UNIX, SnapManager handles all interactions with FlexClone.

Prerequisites, considerations and limitations

Prerequisites

Do the following before using SnapManager:

- ◆ Set up the host system and storage systems. On the host, you must have:
 - ❖ A host operating system and appropriate patches installed.

See the SnapManager & SnapDrive Compatibility Matrix on the NOW site at <http://now.netapp.com/NOW/knowledge/docs/docs.shtml> in the Interoperability section for the latest version and patch information. At this writing, this version supports the following hosts.

Supported Hosts
Red Hat® Enterprise Linux®
SUSE® Linux® Enterprise
Solaris®
IBM® AIX®
HP-UX®

- ❖ Install and configure Oracle. Set up any needed Oracle components, such as RMAN or ASM.
 - ❖ Install NetApp's SnapDrive for UNIX for the host platform, including the products it requires such as the Network Appliance™ Host Attach Kit (kit) or Host Support Kit (kit). Follow the instructions provided with the kit to set up your storage systems to work with the host.
- ◆ Confirm each storage system has:
 - ❖ Data ONTAP installed.
 - ❖ A FlexClone™ license is required to take full advantage of SnapManager with NFS databases. SnapManager uses FlexClone to:
 - Mount backups of NFS databases.
 - Verify backups of NFS databases.
 - Clone NFS databases.
 - Register backups of NFS databases with RMAN (if using RMAN).

FlexClone is part of the Data ONTAP® version supported by SnapManager. As with SnapDrive for UNIX, SnapManager handles all interactions with FlexClone.

- ❖ A SnapRestore license.
- ❖ The correct protocol (FCP, iSCSI, or NFS) installed and licensed.

Host	Protocol
Red Hat® Enterprise Linux®	iSCSI, FCP, NFS
SUSE® Linux® Enterprise	iSCSI, FCP, NFS
Solaris®	iSCSI, FCP, NFS
IBM AIX	NFS
HP-UX	NFS

See the SnapManager & SnapDrive Compatibility Matrix on the NOW site at <http://now.netapp.com/NOW/knowledge/docs/docs.shtml> in the Interoperability section for the latest version information.

Considerations

- ◆ SnapManager is part of the NetApp product set. Use SnapManager as a stand-alone product, or with other NetApp products, such as SnapVault. SnapManager takes advantage of existing NetApp tools and technologies, including snapshots.

SnapManager works with the following:

- ◆ File Storage: Raw, ASM, EXT3, VxFS (VERITAS®)
- ◆ Volume managers:
 - ❖ RedHat Enterprise Linux: Logical Volume Manager (LVM)
 - ❖ SUSE Linux Enterprise: Logical Volume Manager (LVM)
 - ❖ Solaris: VERITAS Volume Manager (VxVM)
 - ❖ AIX: none
 - ❖ HP-UX: none

Topologies

SnapManager supports the same host cluster and storage system cluster configurations that SnapDrive for UNIX and the kit you're using with it support. These include the following topologies:

- ◆ Nonclustered configuration where a single host is connected to a single storage system
- ◆ Any of the topologies involving storage systems running Data ONTAP cluster failover
- ◆ Any of the topologies involving supported host clusters

See the kit documentation for more information about the recommended configurations for your host and storage systems.

Note

Contact your sales representative if you need a SnapManager configuration that is not mentioned in the kit documentation.

Oracle limitations

This release of SnapManager has the following limitations with Oracle:

- ◆ There is no support for Oracle Cluster File System (OCFS).
- ◆ If you are using iSCSI or FCP on Linux, use the Automatic Storage Management support library (ASMLib).

NFS limitations

SnapManager for Oracle runs as root and must be able to access the filesystems containing Oracle data. If using NFS storage, set the anon=0 or Anonymous User ID=0 option on the export for any volumes containing the following:

- ◆ Oracle Data files
- ◆ Redo and Archive Log Files
- ◆ Oracle Home

ASM limitations

If using ASM, be aware of the following ASM restrictions:

- ◆ As recommended by Oracle, make sure you have exactly one partition for each disk that contains the entire disk. You will encounter a SnapManager error if you attempt to back up an ASM disk that has multiple partitions.
- ◆ ASM disks must be raw LUNs. Using files on an NFS or host-side file system as ASM disks is not supported.
- ◆ You must manually register an ASM instance with the listener via `listener.ora`. To your `$ORACLE_HOME/network/admin/listener.ora` add:

```
SID_LIST_LISTENER =
(SID_LIST =
```

```

(SID_DESC =
  (ORACLE_HOME = /u01/app/oracle/10.1.0.2/product/10.1.0/db)
  (SID_NAME = +ASM)
)
)

```

Now restart the both the listener and the ASM instance.

- ◆ Set a profile with RMAN if you want to be able to restore ASM files in all cases. SnapManager needs RMAN to restore the files in the following situations:
 - ❖ The file extents for the file being restored have changed.
 - ❖ The storage system on which the file is stored does not support partial LUN restore.
 - ❖ A full disk group restore is not possible since there are other files on that disk group which are not being restored.
- ◆ When laying out your database, follow the recommendations in the NetApp technical reports TR 3329 (http://www.netapp.com/tech_library/3329.html) and TR 3349 (http://www.netapp.com/tech_library/3349.html). These reports provide information on how to lay out the LUNs for your ASM disk groups.

Other limitations

- ◆ SnapManager cannot automatically archive the backups to, or restore archived backups from, another storage system in this release. The snapshots exist on the same storage system as the primary data.
- ◆ SnapManager for Oracle supports only one file system on an LVM Volume Group or VERITAS Disk Group. If SAN LUNs are used in combination with LVM or VxVM and a host file system, each file system must reside in its own independent Volume Group or Disk Group.

About this chapter

This chapter explains the SnapManager for Oracle interface options and ways to access each interface. It contains the following topics:

- ◆ “[Interface choices](#)” on page 34
- ◆ “[Using the SnapManager for Oracle GUI](#)” on page 35

Interface choices

Use SnapManager for Oracle in either of the following ways:

- ◆ From the command line of your database host, by issuing commands from the CLI. Use a single command to perform each operation. The CLI provides the ability to invoke SnapManager for Oracle from scripts.
- ◆ From the GUI on a Windows host in the same network as your database host. The GUI provides a simple tool, with easy-to-use wizards to help you perform all SnapManager for Oracle tasks.

Starting the SnapManager CLI

Use SnapManager for Oracle from the CLI by entering a command at the prompt on your UNIX host. The CLI commands all start with `smo` (SnapManager for Oracle). For example, you have commands such as `smo repository create` or `smo backup create`.

For a list of all the SnapManager CLI commands and an explanation of their options and arguments, see “[SnapManager Command Reference](#)” on page 133.

If you have chosen to use the CLI, skip over the next sections about Starting and Using the SnapManager for Oracle GUI. To get started installing and using SnapManager, go to “[Installing or Upgrading SnapManager](#)” on page 41.

Starting the SnapManager GUI

To start the SnapManager GUI from the Windows host do one of the following:

- ◆ Double-click the SMO GUI 2.0 shortcut on your desktop.
- ◆ Select **Start > Programs > NetApp > SnapManager for Oracle GUI 2.0 > Start SMO**.

Note

You cannot run the SnapManager CLI on the Windows host.

Using the SnapManager for Oracle GUI

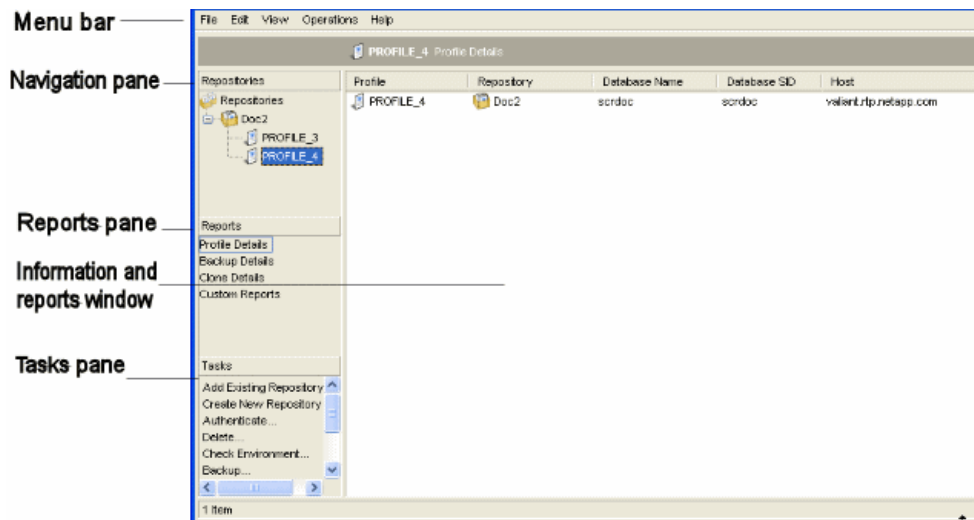
In addition to performing operations, the GUI provides a central method for managing a backup's resources. There are several ways to perform SnapManager operations from the GUI.

- ◆ Select the operation from the pull-down menus in the menu bar. For example, to create a profile, select Operations > Repository > Create Profile.
- ◆ Click on the operation in the Tasks pane of the GUI screen.
- ◆ Select the object (repository, profile, or backup) that the operation applies to and right-click on the name. A menu appears with a list of operations you can perform. Click the operation.

If the operation contains multiple steps, such as creating a profile or backing up a database, SnapManager starts a wizard to step you through the procedure.

Navigating the main GUI screen

The following screen shows the basic operations you can perform from the GUI. The main GUI screen is divided so that operations and tasks appear in the panels on the left of the screen while reports and operation status appear in the rest of the screen.



Perform operations using the pull-down menus from the Menu bar at the top of the screen.

Access the SnapManager for Oracle GUI help documentation from the toolbar.

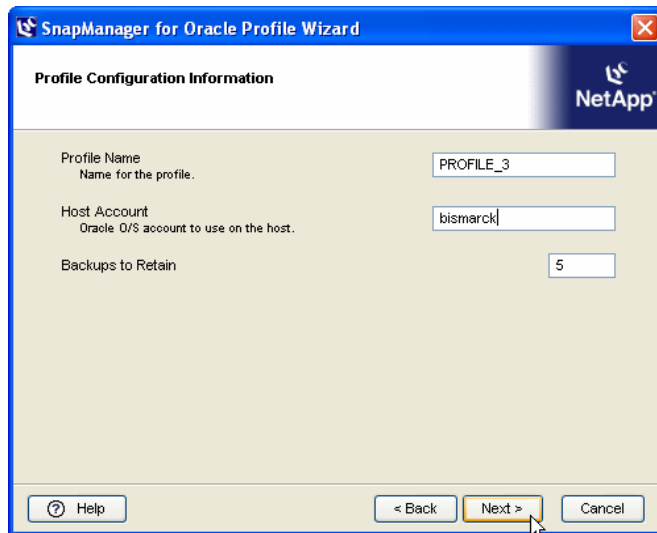
The panes on the left of the screen fall into the following categories:

- ◆ **Navigation pane.** This pane uses a tree structure to display information on the repositories, profiles, backups, and clones. The initial display shows the repositories that are available. Click a repository name and SnapManager displays a list of the profiles contained in the repository. Click a profile name and SnapManager displays the backups of the database in the profile. If this is the first time you've clicked on that profile during this session, SnapManager displays authentication pop-up window before it displays the backups. Provide the correct authentication information for the database.
To create a backup, first select the repository and then select the profile for the database from that repository. To restore a backup, select the repository, then chose the profile for the database, and finally, select the backup to use. SnapManager lets you retain as many backups as you want, providing there is enough space for them.
- ◆ **Reports pane.** This pane contains a list of reports. The list of reports changes depending on what is selected in the Navigation pane. There are different reports for repositories, profiles, and backups. To see a report, click on the report name. SnapManager displays the report in the main section of the screen.
- ◆ **Tasks pane.** This pane lists the operations you can perform. The list changes depending on what is selected in the Navigation pane and Reports pane.
When you click on a task, SnapManager starts the wizard to step you through the operation.
For example, select a repository in the Navigation panel, and choose the task to create a new profile. SnapManager starts the Profile Wizard.

The main body of the screen is the Information and Reports window. SnapManager displays operation status in this section. It also displays any reports you select in this area.

Wizard screens

The wizard screens are titled with the name of the task you are performing. For example, the SnapManager for Oracle Profile Wizard opens when creating a profile. The following is an example of a wizard screen.



The image shows a Windows-style dialog box titled "SnapManager for Oracle Profile Wizard". The title bar is blue with a NetApp logo on the left and a close button on the right. The main area has a white background with the title "Profile Configuration Information" in bold. Below this, there are three configuration fields: "Profile Name" with the value "PROFILE_3", "Host Account" with the value "bismarck", and "Backups to Retain" with the value "5". At the bottom, there are three buttons: "Help" (with a question mark icon), "< Back", and "Next >" (which is highlighted with a yellow border and a mouse cursor). A "Cancel" button is also present to the right of the "Next >" button.

SnapManager for Oracle Profile Wizard

Profile Configuration Information

Profile Name
Name for the profile. PROFILE_3

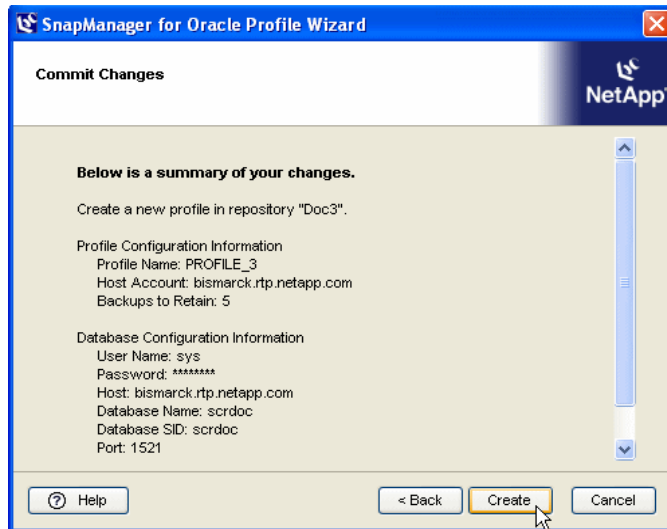
Host Account
Oracle O/S account to use on the host. bismarck

Backups to Retain 5

Help < Back Next > Cancel

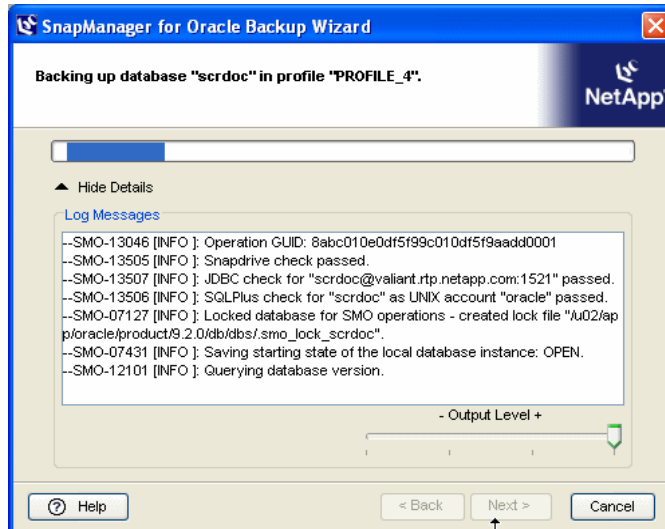
Committing to changes

After entering any configuration information in the GUI, the Commit Changes window opens, asking you to review the information for the task being executed. If the information is correct, execute the task. Click the Back button to go back to a previous screen to enter correct information.

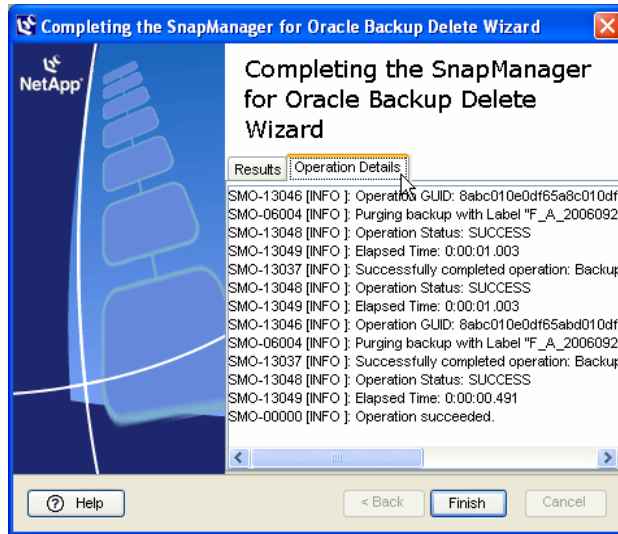


Viewing operation details

In any wizard, when you commit to the changes and execute the operation, there is a status page with the log messages.

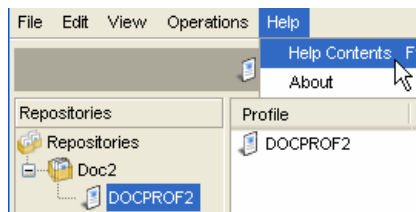


Once the operation is complete, the Completing the SnapManager task window opens and you can click on the Operation Details tab to view the log for that operation. If the operation was successful, just click Finish to close the window and return to the main screen.



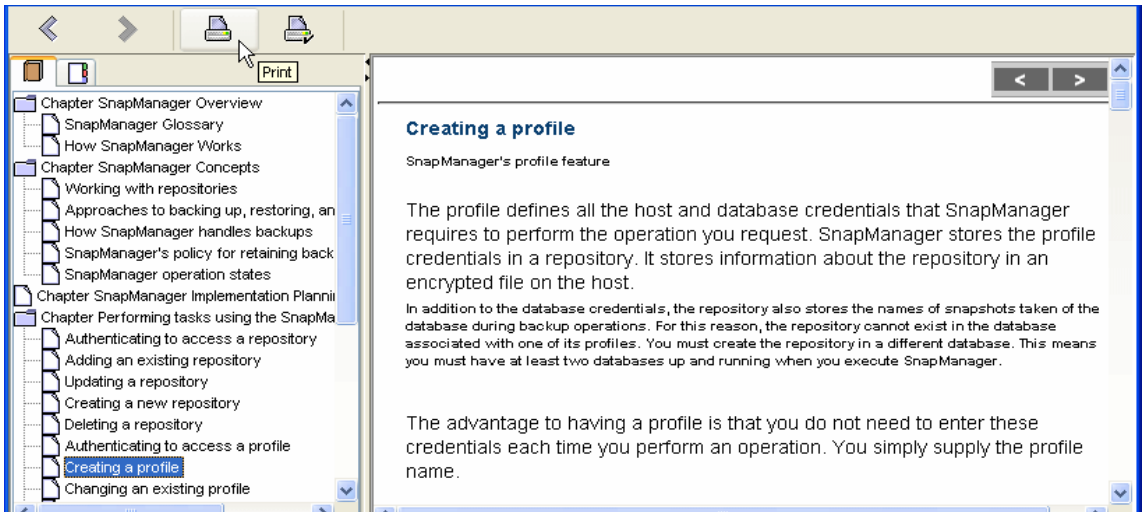
Accessing GUI help

In the main GUI screen, click Help in the menu bar. Select Help Contents to view the complete set of help topics for the GUI.



From this screen:

- ◆ Navigate through all of the topics in this help window by using the topic table of contents in the left pane.
- ◆ Print out the help topics by clicking the printer icon at the top of the help window.



Accessing context-sensitive help

When performing a task, click Help in the lower left corner of any screen of a task wizard, and the appropriate context sensitive help window opens to guide you through the process.

For example, click the Help button in the Profile Create Wizard and the help window for creating a profile appears.

About this chapter

This chapter explains how to install or upgrade your version of SnapManager for Oracle from downloading the software to the point where you can get started using SnapManager for Oracle.

This topics in this chapter are:

- ◆ [“Installing and upgrading SnapManager tasks”](#) on page 42
- ◆ [“Downloading the SnapManager software package”](#) on page 44
- ◆ [“Installing or upgrading the SnapManager software packages”](#) on page 46
- ◆ [“Verifying, starting and stopping the server”](#) on page 49

Installing and upgrading SnapManager tasks

If your system is set up correctly, SnapManager is easy to install and start running. From the NOW site, download and install the software package for your host operating system. To use the SnapManager GUI, also download and install the GUI package on a Windows system.

Before installing SnapManager, make sure that your system is set up correctly and has all the prerequisites installed.

The following is a high-level list of the tasks involved in preparing for and installing SnapManager.

Step	Action	Details
Preparing to install the SnapManager software package		
1	Make sure your system meets the SnapManager requirements. Host prerequisites: ◆ Snap Drive for UNIX and its requirements. Storage system prerequisites: ◆ On the storage system, have Data ONTAP installed and running. If you haven't set up your system yet, see "SnapManager Deployment Considerations" on page 27. You should also have the following licenses: ◆ FlexClone (if using NFS) Note _____ SnapManager uses the FlexClone feature to clone NFS databases and register NFS database backups with RMAN. _____ ◆ SnapRestore	See "Downloading the SnapManager software package" on page 44. There is also information about the documentation available for these NetApp products in "Where to go for more information" on page 131.

Step	Action	Details
2	Download a copy of the SnapManager software package as well as a copy of the GUI, if using it. You can also get a copy of the software from the SnapManager CD.	See the section “ Download the SnapManager software from NOW ” on page 44.
Installing or Upgrading the SnapManager software package		
3	Follow the instructions to install SnapManager on the database host.	For all platforms, see the section “ Installing or upgrading SnapManager files on a database host ” on page 46.
4	(Optional) Install the SnapManager GUI on a Windows system.	See the section “ Installing or upgrading the GUI on a Windows system (Optional) ” on page 47.
Preparing to use SnapManager		
5	On the database host, start the SnapManager server component using the <code>smo_server start</code> command.	See the section “ Starting the server procedure ” on page 49.
6	If installing SnapManager for the first time on this host environment, create a new repository for the profiles to use for the databases. If upgrading SnapManager, update your existing repositories.	See the section “Updating and creating repositories” on page 52. If using the GUI, refer to the GUI online help.
7	Create profiles for the databases you want to back up and restore.	See the section “ Creating a profile ” on page 63.
8	Check to make sure all the required components are installed and working with SnapManager.	“ Verifying the environment ” on page 62.

Downloading the SnapManager software package

Check requirements for SnapManager and its prerequisites

The most up-to-date information about SnapManager and the other NetApp products it uses is on the NOW site. For information on:

- ◆ SnapManager requirements, go to the SnapManager Compatibility Matrix, located in the Interoperability section on the page <http://now.netapp.com/NOW/knowledge/docs/docs.shtml>. To get to this page, follow steps 1 through 4 in the section “[Download the SnapManager software from NOW](#)” on page 44.
- ◆ SnapDrive for UNIX requirements, check the online *SnapDrive for UNIX Compatibility Matrix* and the *Configuration Guide for NetApp's FCP and iSCSI Products*. They are both available at <http://now.netapp.com/NOW/knowledge/docs/docs.shtml>.

Where to find the software

NetApp provides the SnapManager software on both a SnapManager CD and the NOW site. Both the CD and the NOW site contain the software to install on the UNIX host and on the Windows machine where you will run the GUI.

To install the software:

1. Download the host software file for the host operating system from the NOW site or get a copy from the CD.
2. Install the software.
3. If you want to use the GUI, also get a copy of the file containing the GUI software, either by downloading it from the NOW site or from the CD.

Make sure that you have SnapDrive for UNIX installed on your UNIX host. If you do not have this product, download it from the appropriate location on the NOW site. Contact your sales representative to arrange to purchase this product.

Download the SnapManager software from NOW

To download the SnapManager software from NOW, complete the following steps.

Step	Action
1	Log in to http://now.netapp.com. In the fields on the right side of the page, complete the following steps: 1. Enter your user name and password. 2. From the Select Start Page list box, click Download Software. 3. Click the Login button.
2	Go to the SnapManager product row of the Software Download table and select Oracle from the Select Platform drop-down list.
3	Click Go!
4	Follow the prompts until you reach the SnapManager Description Page. Review the information on this page. At the bottom of this page, click Continue to go to the Software Download page. Downloading other NetApp products: If you need to download SnapDrive for UNIX, go to the correct row in the Software Download page for the product. Then follow the links to the product's Software Description page. Download the documentation from the Software Description page. For more information on downloading this product, see the Software Download page for SnapManager.
5	Download the software file for your host platform. If using the GUI, you must also download the GUI installer.

Installing or upgrading the SnapManager software packages

Installing or upgrading SnapManager files on a database host

To install the SnapManager files on your database host, complete the following steps.

Step	Action
1	Log in as root.
2	Change to the directory on your database host where you put the software you downloaded from the NOW site or copied from the CD.
3	Use the following command to change permissions on the downloaded file so it is executable: <pre># chmod 544 netapp.smo*</pre>
4	Use the following command to start the installation of the software on the host. For Linux, use: <pre># ./netapp.smo.linux-2.0.0.bin</pre> For Solaris, use: <pre># ./netapp.smo.sunos-2.0.0.bin</pre> For IBM AIX, use: <pre># ./netapp.smo.aix-2.0.0.bin</pre> For HP-UX, use: <pre># ./netapp.smo.hpux-2.0.0.bin</pre> Result: The command installs the software in the: <pre>/opt/NTAPsmo</pre> directory for Solaris <pre>/opt/NetApp/smo</pre> directory for all other platforms
5	After the Introduction text, press Enter to continue. If this is an upgrade, you receive the message “Existing SnapManager For Oracle Detected.” Press Enter to continue.
6	At the prompt for operating system user, press Enter to accept the default value. For a new installation, the default value is <i>oracle</i> .
7	At the prompt for operating system group, press Enter to accept the default value. For a new installation, the default value is <i>dba</i> .

Step	Action
8	At the Configuration Summary screen, press Enter to continue. If this is a new installation, the software is installed and includes the required JRE, and then runs the <code>smo_setup</code> script. If this is an upgrade, the message “Begin Uninstall of Existing SnapManager for Oracle” appears. Press Enter to continue. The installer uninstalls the existing software and then installs the new software.
9	At the message Installation Complete, press Enter to exit the installer.
10	Start the server by using the following command syntax: <code>smo_server start</code>
11	If upgrading from a previous SnapManager version that used the NetApp Host Agent, uninstall the Host Agent software, if the Host Agent is not being used for any other purpose. The new version of SnapManager for Oracle no longer uses the Host Agent. However, the Host Agent may be still be used for other purposes.

If you are not planning to install or use the GUI, skip the next section and see the section “[Verifying, starting and stopping the server](#)” on page 49.

Installing or upgrading the GUI on a Windows system (Optional)

To install the SnapManager GUI complete the following steps.

Step	Action
1	Log on to the Windows system with an account that has sufficient privileges to install programs in C:\Program Files.
2	Download the <code>netapp.smo.windows-2.0.0.exe</code> file containing the SnapManager GUI to C:\Program Files.
3	Double click the downloaded file.
4	Click OK if a dialog box opens with the message, “The publisher could not be verified. Are you sure you want to run this software?”
5	The Introduction screen opens. Click Next .

Step	Action
6	At the next screen, click Next to accept the default location for the Installation Folder. The default location is: C:\Program Files\NetApp\smo-2.0.0.
7	At the Choose Icon Availability screen, click next to provide a desktop SnapManager for Oracle icon.
8	At the Pre-Installation Summary screen, click Install . The installer installs both the SMO Client and the required JRE.
9	At the Install Complete screen, click Done .

Verifying, starting and stopping the server

Verifying the status of the server procedure

To verify the status of the server, use the following command as root:

```
smo_server status
```

If the server is already running, go to “[Updating and Creating Repositories](#)” on page 51. If it is not running, start the server using the following procedure.

Starting the server procedure

Before you can use the GUI or the CLI to initiate SnapManager for Oracle commands, the SnapManager for Oracle server must be running. To start the server, use the following command as root:

```
smo_server start
```

After starting the server, see section “[Information required to update or create a repository](#)” on page 52 to continue configuring SnapManager for Oracle.

Stopping the server

To stop the server, use the following command as root:

```
smo_server stop
```


About this chapter

This chapter provides information about creating and updating your repositories.

This chapter includes the following topics:

- ◆ [“Information required to update or create a repository”](#) on page 52
- ◆ [“About updating repositories”](#) on page 53
- ◆ [“Updating a repository procedure”](#) on page 53
- ◆ [“Creating a repository procedure”](#) on page 54

Updating and creating repositories

Information required to update or create a repository

The basic steps to update or create a repository are as follows:

1. Identify an existing database or create a new one to associate with the repository.

Attention

It is highly recommended that you backup existing repository databases prior to initiating the update process. A failure during the update process could leave your repository database in an inconsistent state.

Note

After you update the repository, you cannot use the repository with an older version of SnapManager for Oracle.

2. To create the repository, SnapManager for Oracle requires a minimum of 4 K block size for the tablespace into which it is installed. Check the block size using:

```
select TABLESPACE_NAME, BLOCK_SIZE from dba_tablespaces
```

3. Specify a name for the repository. This can be up to 30 characters.
4. Supply the information required:
 - ❖ Enter the repository name. Use either the global name or the SID for the repository.
 - ❖ Specify the user name/password required to connect to the database.
 - ❖ Specify the host where the database resides. Use either the host name or the IP address.
 - ❖ Specify the port number SnapManager must use to connect to the repository. Normally, this is 1521 (Oracle's default listener port).
 - ❖ If a repository schema already exists, specify the force flag to overwrite it with the new repository.

Attention

All data within the existing repository schema will be overwritten when you force the creation of the new repository.

When you update or create a new repository, SnapManager for Oracle names the repository using this convention:

username@DBHOST:REPO_DBNAME

About updating repositories

If you have no existing repositories, see [“Creating a repository procedure”](#) on page 54.

Attention

It is highly recommended that you backup existing repository databases prior to initiating the update process. A failure during the update process could leave your repository database in an inconsistent state.

You must update your existing repositories when upgrading to this version of SnapManager for Oracle. This updates your database schema and objects so you can access them using the new version of SnapManager for Oracle.

When you update repositories, SnapManager issues an update confirmation prompt and recommends you backup any existing repository databases before updating. If you have already backed up the repository database, you can use the `-noprompt` option. You do not receive the confirmation prompt when you use this option.

Note

After you update the repository, you cannot use the repository with an older version of SnapManager for Oracle.

Updating a repository procedure

To update the repository, use the `smo repository update` command.

```
smo repository update
  {-repository -dbname db_name -login -username username -password
  password -host host -port port_number}
  [-sid sid]
  [-noprompt]
  [-quiet | -verbose]
```

Example: The following command line updates an existing repository.

```
# smo repository update -repository -dbname DB2 -login -username
user2 -password aloha2 -host host2 -port 1521
```

Creating a repository procedure

To create a repository, use the `smo repository create` command. It has the following general format:

```
smo repository create
  {-repository -dbname db_name -login -username username -password
  password -host host -port port_number}
  [-sid sid]
  [-force]
  [-noprompt]
  [-quiet | -verbose]
```

For details on the options and arguments used in this command, see “[The smo repository create command](#)” on page 182.

Example: The following command line creates a repository.

```
# smo repository create -repository -dbname DB1 -login -username
user1 -password high -host host1 -port 1521
```

About this chapter

This chapter provides information about how to get started using SnapManager for Oracle.

This chapter includes the following topics:

- ◆ [“Before you begin”](#) on page 56
- ◆ [“Example procedure”](#) on page 58
 - [“Task 1: Verify the environment”](#) on page 58
 - [“Task 2: Create a profile”](#) on page 58
 - [“Task 3: Verify the profile configuration”](#) on page 59
 - [“Task 4: Back up your database”](#) on page 59
 - [“Task 5: Verify the backup”](#) on page 59
 - [“Task 6: Restore the backup”](#) on page 60
 - [“Task 7: Create a clone from the backup”](#) on page 60

Before you begin

Review of important profile concepts

Consider these basic profile concepts:

- ◆ SnapManager works with profiles you set up. Each profile contains the credentials for the database associated with the profile. The credentials enable SnapManager to connect to the database and work with it. You must set up a profile for each database you want to use before you can back it up.
- ◆ Profile information is stored in a repository. The repository contains both the profile information for the database and information about the snapshots that provided the backup of the database. (The actual snapshots are stored on the storage system.) When you perform an operation on a database, you must select the profile from the repository. Because the repository contains the details on the backup of the database, it must exist in a separate database. It cannot exist in the same database as the one you are backing up.
- ◆ The backups are point-in-time copies of the database using NetApp's snapshot technology. The names of these snapshots are stored in the repository containing the profile for that database.

SnapManager's profile feature

To simplify working with the databases, SnapManager has you set up a profile for each database. The advantage to having a profile is that you do not need to enter these credentials each time you perform an operation. You simply supply the profile name.

The profile defines the host and database credentials that SnapManager requires in order to perform the operation you request. SnapManager stores the profile credentials in a repository. It stores information about the repository in an encrypted file on the database host.

Note

In addition to the database credentials, the repository also stores the names of snapshots taken of the database during backup operations. For this reason, the repository cannot exist in the database associated with one of its profiles. You must create the repository in a different database. This means you must have at least two databases up and running when you execute SnapManager.

Review of prior activities

At this point you should have performed the following actions.

Prerequisite Action	Where to find information
Download and install the SnapManager for Oracle software	“Installing or Upgrading SnapManager” on page 41
Start the server	“Verifying, starting and stopping the server” on page 49
Update existing repositories to make the database schema and objects accessible, or create a new repository	“About updating repositories” on page 53 and “Updating a repository procedure” on page 53 “Information required to update or create a repository” on page 52 and “Creating a repository procedure” on page 54
Determine which interface you will use	“SnapManager Interfaces” on page 33. If using the GUI, open the GUI and refer to the online help.

Example procedure

Task 1: Verify the environment

Once you've installed SnapManager, verify that all the components for your system are set up correctly. Use the `smo system verify` command to verify the environment.

Example:

```
# smo system verify
```

Output:

```
SMO-13505 [INFO ]: Snapdrive verify passed.  
SMO-13037 [INFO ]: Successfully completed operation: System Verify  
SMO-13049 [INFO ]: Elapsed Time: 0:00:00.559  
Operation Id [N4f4e910004b36cfecee74c710de02e44] succeeded.
```

Task 2: Create a profile

Use the `smo profile create` command to create a profile from the CLI.

Once you have created a profile you can use the `smo profile show` command display information about a profile. To see a list of profiles for that repository, use the `smo profile list` command

For details on the options and arguments used in these commands, see [“The smo profile create command”](#) on page 165, the [“The smo profile list command”](#) on page 173, and the [“The smo profile show command”](#) on page 174.

Examples:

The following command line sets up a profile that uses ASM.

```
# smo profile create -profile SMOKE_SMOASM_BISMARCK -repository  
-dbname SMOREPO -login -username bismarck_smoke_user -password  
netapp -port 1521 -host bismarck -database -dbname SMOASM -sid  
SMOASM -login -username sys -password oracle -port 1521 -host  
bismarck -osaccount oracle -asm -host valiant -login -username  
asmuser -password asmpassword -port 1522 -sid asmsmo
```

Output:

```
Operation Id [8abc01ec0e79259b010e7925a2790005] succeeded.
```

This example adds the option to retain three backup copies. (The retain option applies only to successful backups that have not been used to create a clone. Regardless of the value for retain, SnapManager does not automatically remove backups that were used to create a clone.)

```
# smo profile create -profile SMOKE_SMO_BISMARCK -repository
-dbname SMOREPO -login -username bismarck_smoke_user -password
netapp -port 1521 -host bismarck -database -dbname SMOASM -sid
SMOASM -login -username sys -password oracle -port 1521 -host
bismarck -retain 3 -osaccount oracle
```

Task 3: Verify the profile configuration

Use the `smo profile verify` command to verify the profile just created. For details on the options and arguments used in this command, see [“The smo profile verify command”](#) on page 181.

```
# smo profile verify -profile SMOKE_SMO_BISMARCK
```

Output:

```
Operation Id [N56aa6482158b711c86cb499aa137bb03] succeeded.
```

Task 4: Back up your database

Use the `smo backup create` command to create a backup.

For details on the options and arguments used in this command, see [“The smo backup create command”](#) on page 137.

Example: The following command line creates a backup using a label.

```
# smo backup create -full -offline -profile SMOKE_SMO_BISMARCK
-label smoke_full_backup -force
```

Output:

```
Operation Id [8abc01ec0e79356d010e793581f70001] succeeded.
```

Task 5: Verify the backup

To verify a backup, use the `smo backup verify` command. For details on the options and arguments used in this command, see [“Using the smo backup verify command”](#) on page 153.

Example: The following command line verifies the backup.

```
# smo backup verify -profile SMOKE_SMO_BISMARCK -label
smoke_full_backup -quiet
```

Output:

```
DBVERIFY - Verification starting : FILE =
+SMO_1_1161675083835/smoasm/datafile/smoke.277.582482539
...
```

Task 6: Restore the backup

Use the `smo backup restore` command to restore a backup.

For details on the options and arguments used in this command, see “[The smo backup restore and recovery command](#)” on page 148.

Example: This example restores the entire backup that is part of the profile. It also recovers the database to the last commit.

```
# smo backup restore -complete -profile SMOKE_SMO_BISMARCK -label  
smoke_full_backup -force -recover -alllogs
```

Output:

```
Operation Id [8abc01ec0e79442c010e79444cf90001] succeeded.
```

Task 7: Create a clone from the backup

Use the `smo clone create` command to clone a backup of a database instance. For details on the options and arguments used in this command, see “[The smo clone create command](#)” on page 155.

Example: This example explains how to create a clone from a backup.

1. Create a clone specification file or use an existing clone specification.

Note

Clone specifications differ from the clone maps in previous releases of SnapManager for Oracle and are not backwards compatible. You cannot use a clone map to create a clone with this version. You must create a clone specification by following the instructions in “[Creating a clone specification](#)” on page 80.

2. The following example creates a clone from a backup using an existing clone specification, called `SMO_bismarck.xml`, which is located on the host.

```
# smo clone create -newsid CLONE -clonespec  
/opt/NetApp/smo/clonespecs/SMO_bismarck.xml -profile  
SMOKE_SMO_BISMARCK -label smoke_full_backup
```

Output:

```
Operation Id [8abc01ec0e794e3f010e794e6e9b0001] succeeded.
```

About this chapter

This chapter provides information about verifying your environment is set up correctly, setting up a database profile, and verifying the profile.

This chapter includes the following topics:

- ◆ [“Verifying the environment”](#) on page 62
- ◆ [“Creating a profile”](#) on page 63
- ◆ [“Verifying the profile”](#) on page 66

Verifying the environment

Once you've installed SnapManager, you can verify that all the components for your system are set up correctly. This operation confirms that:

- ◆ SnapDrive is installed and can be run from the root account.

Running `smo system verify` tells SnapManager to verify that:

- ◆ The database is accessible through the listener port.
- ◆ SQLPlus and RMAN can be run from this Oracle account.
- ◆ The ASM instance is accessible through the listener port (only if you are using ASM).
- ◆ The RMAN catalog is available (this only needs to be available if you are using RMAN).

The command has the following general format:

```
smo system verify [-quiet] [-verbose]
```

For details on the options and arguments used in this command, see “[The smo system verify command](#)” on page 189.

Creating a profile

Information required to create a profile

This section takes you through the basic steps. For a syntax example, see “[Creating a profile procedure](#)” on page 64.

The basic steps for creating or modifying a profile are as follows:

1. Specify a name for the profile. This can be up to 30 characters. It must be unique within the repository that contains the profile and within the server that accesses the profile.
2. Supply the information for the repository that will contain the profile. You must have already created the repository before you can create a profile.
 - ❖ Enter the repository name. Use the global database name for the repository.
 - ❖ Specify the user name/password required to connect to the database.
 - ❖ Specify the host where the repository resides. You can use either the host name or the IP address.
 - ❖ Specify the port number SnapManager must use to connect to the repository. Normally, this is 1521 (Oracle’s default listener port).
3. Supply the information required by the database associated with the profile.
 - ❖ Enter the database name.
 - ❖ Specify the host where the database resides. You can use either the host name or the IP address.
 - ❖ Specify the user name/password required to connect to the database.
 - ❖ Specify the port number SnapManager must use to connect to the database. Normally, this is 1521 (Oracle’s default listener port).
 - ❖ If necessary, specify the SID. By default, SnapManager uses the database name as the SID. If you are using RAC, specify the SID used for the instance on the corresponding host in the cluster.
4. If using ASM, enter the information required.
 - ❖ Specify the user name/password required to connect to ASM.
 - ❖ Specify the port number SnapManager must use to connect to ASM. Normally, this is 1521 (Oracle’s listener port).
 - ❖ Specify the SID. If you are using RAC, specify the SID used for the instance on the corresponding host in the cluster.

5. If using the RMAN, specify that you will use either:
 - ❖ Control files or the RMAN catalog.
 - ❖ To use the RMAN catalog, you must provide:
 - * The user name/password required to connect to the RMAN catalog.
 - * The tnsname connection name (this is defined in the `tnsname.ora` file).
6. Enter the name of the Oracle database UNIX user account (osaccount).
7. Enter the number of successful backups that you want SnapManager to retain for this database. This option applies only to successful backups that have not been used to create a clone. For more information on how SnapManager retains backups, see [“SnapManager’s policy for retaining backups”](#) on page 23.

Creating a profile procedure

To create a profile from the CLI, use the `smo profile create` command. It has the following general format:

```
smo profile create
-profile
{-repository -dbname repo_dbname -host repo_host -login
-password repo_password -username username -port repo_port}
{-database -dbname dbname -host dbhost -login -password db_password
-username db_username -port db_port [-sid db_sid]}
{-osaccount account_name}
[-rman {-tnsname tnsname -login -password rman_password -username
rman_username} | -controlfile ]
[-asm -host asm_host -login -password asm_password -username
asm_username -port asm_port -sid asm_sid ]
[-retain number]
[-quiet | -verbose]
```

Once you have created a profile you can use the `smo profile show` command to display information about a profile. To see a list of profiles for that installation, use the `smo profile list` command

For details on the options and arguments used in these commands, see [“The smo profile create command”](#) on page 165, the [“The smo profile list command”](#) on page 173, and the [“The smo profile show command”](#) on page 174.

Examples:

The following command line sets up a profile that uses ASM and RMAN.

```
# smo profile create -profile SMOKE_SMOASM_BISMARCK -repository
-dbname SMOREPO -login -username bismarck_smoke_user -password
netapp -port 1521 -host bismarck -database -dbname SMOASM -sid
SMOASM -login -username sys -password oracle -port 1521 -host
bismarck -osaccount oracle -asm -host valiant -login -username
asmuser -password asmpassword -port 1522 -sid asmsmo -rman
-username rmanuser -password rmanpassword
```

This is the same example without the ASM and RMAN information. This command line adds the option to retain three backup copies. (The retain option applies only to successful backups that have not been used to create a clone. Regardless of the value for retain, SnapManager does not automatically remove backups that were used to create a clone.)

```
# smo profile create -profile SMOKE_SMO_BISMARCK -repository
-dbname SMOREPO -login -username bismarck_smoke_user -password
netapp -port 1521 -host bismarck -database -dbname SMOASM -sid
SMOASM -login -username sys -password oracle -port 1521 -host
bismarck -retain 3 -osaccount oracle
```

Output:

```
Operation Id [8abc01ec0e78ebda010e78ebe6a40005] succeeded.
```

Verifying the profile

Verifying the profile procedure

Enter the name of profile you created previously. SnapManager checks the environment for this profile. This profile defines all the credentials that SnapManager requires when working with a database and confirms that the database included in this profile is accessible.

To verify the profile, use this command:

```
smo profile verify -profile profile_name [-quiet | -verbose]
```

Example:

```
# smo profile verify -profile SMOKE_SMO_BISMARCK
```

Output:

```
Operation Id [N56aa6482158b711c86cb499aa137bb03] succeeded.
```

More information about managing profiles

For information about changing existing profiles or deleting profiles, see the chapter “[Managing SnapManager Operations](#)” on page 93.

About this chapter

This chapter describes how to use SnapManager to backup and restore databases. It includes the following topics:

- ◆ [“Creating database backups”](#) on page 68
- ◆ [“Verifying a backup”](#) on page 71
- ◆ [“Restoring and recovering a database backup”](#) on page 73

Creating database backups

SnapManager lets you create backups of entire databases and portions of databases. The backups are virtual copies of the database instance and are stored on the same physical medium as the database.

SnapManager's backups consist of the snapshots taken at the point in time when the database instance was backed up. It uses snapshots to create the backups. You can then use the backups to restore and recover the database information at a later time.

For more information on how SnapManager handles backing up information, see [“Approaches to backing up, restoring, and recovering data”](#) on page 14.

General steps for backing up databases

The basic steps for backing up data are as follows:

1. Specify the type of backup you want to perform:
 - ❖ Online. The database is mounted and running (the Oracle status is OPEN), and archive log mode is enabled.
 - ❖ Offline. The database is mounted, but not available to users (the Oracle status is either MOUNTED, STARTED, or SHUTDOWN).
 - ❖ Auto. SnapManager determines whether the database is online or offline and then performs the appropriate backup.
2. (Optional) Specify the verify option if you want SnapManager to verify the backup using the Oracle utility `dbv` after the backup completes. When you select this option, SnapManager checks the block structure of the backup to make sure it is in a valid format for Oracle.

Note

The verification might take several minutes. If you prefer to perform this operation at a later time, you can use the `smo backup verify` operation.

3. Specify the details for the backup:
 - ❖ The name of the profile associated with the database
 - ❖ Whether this is a full or partial backup. If you specify a full backup, SnapManager backs up all the data files. If you specify a partial backup, SnapManager only backs up the tablespace or data files that you specify. You must enter these names in a space delimited list.

Note

If you are performing a full, offline backup, SnapManager backs up the redo logs. SnapManager always backs up the archive logs and the control files.

- ❖ (Optional) Enter a label (i.e., an easily identified name) for the backup. The label name must be unique within that profile. Use alphanumeric characters, underscores, and hyphens, as long as the first character in the name is not a hyphen.

Tip: To make identifying the backup easier, it is recommended that you supply a label for this backup.

If you do not supply a label, SnapManager creates a default label that has the format *scope_type_date* where:

- *scope* is either F to indicate a full backup or P to indicate a partial backup.

- *type* is either C to indicate an offline (cold) backup or H to indicate an online (hot) backup.

- *date* is the year, month, day, and time of the backup.

For example, if you performed a full backup with the database offline on Jan. 16, 2006, at 5:45:16 p.m. Eastern standard time, SnapManager would create the label F_C_20060116174516EST.

- ❖ (Optional) Whether you want to enter a comment that describes the backup. For example, you might enter ‘Production test’ so that you can quickly identify the contents of the backup.
4. (Optional) Enter the `-force` option if you want SnapManager to force the database to change its state if it is not in the state you requested for the backup operation.
 5. (Optional) Specify how you want SnapManager to handle messages when executing the command from the CLI. For more information on these options, see [“Common options for handling messages shown on the console”](#) on page 136.

Creating a backup procedure

To create a backup, use the `smo backup create` command. It has the following general format:

```
smo backup create -online|-offline|-auto
                 -full|{-files list}| {-tablespaces list}
                 {-profile}
                 [-verify] [{-label name}]
                 [-comment comment]
                 [-force] [-quiet | -verbose]
```

For details on the options and arguments used in this command, see “[The smo backup create command](#)” on page 137.

Examples:

This is an example of a partial offline backup of tablespaces.

```
# smo backup create -offline -tablespaces USERS -profile  
SMOKE_NFSTEST_BISMARCK -label smoke_offline_tablespaces -force
```

Output:

```
Operation Id [8abc01ec0e78ecb5010e78ecc2940001] succeeded.
```

This is an example of a full online backup.

```
# smo backup create -online -full -profile SMOKE_NFSTEST_BISMARCK -  
label smoke_online_full -force
```

Output:

```
Operation Id [8abc01ec0e78ee9e010e78eeb75f0001] succeeded.
```

This is an example of a partial online backup of files.

```
# smo backup create -online -files  
/mnt/triggerfish/bt_bismarck_nfs_smoke/datafiles/users01.dbf -  
profile SMOKE_NFSTEST_BISMARCK -label smoke_online_files -force
```

Output:

```
Operation Id [8abc01ec0e78ef7a010e78ef8a280001] succeeded.
```

Verifying a backup

Verify that the backup is valid

You can use SnapManager's verify feature to confirm that physical blocks in the backup of the database have not been corrupted. This feature invokes Oracle's `dbv` command for each data file in the backup.

SnapManager lets you perform the verify operation at any time that is convenient for you and the users on your system. Have SnapManager perform the verification immediately after making the backup, or you can verify the operation at a later time.

General steps for verifying a backup

The basic steps for verifying a backup are as follows:

1. Specify the name of the backup. This can be either:
 - ❖ The label you supplied when you created the backup
 - ❖ The GUID that SnapManager generated when you created the backup. (For more information about the GUID, see [“General notes about the commands”](#) on page 134.)
2. Specify the name of the profile from which the backup was created.
3. (Optional) Specify how you want SnapManager to handle messages when executing the command from the CLI. For more information on these options, see [“Common options for handling messages shown on the console”](#) on page 136.

Verifying a backup procedure

To verify a backup from the CLI, use the SMO verify command. It has the following general format:

```
smo backup verify {-id guid | -label label_name}  
                  -profile profile_name [-quiet | -verbose] [-force]
```

For details on the options and arguments used in this command, see [“Using the smo backup verify command”](#) on page 153.

Example: The following command line verifies the backup labeled `smoke_full_backup`.

```
# smo backup verify -profile SMOKE_NFSTEST_BISMARCK -label  
smoke_full_backup -quiet
```

Output:

```
DBVERIFY - Verification starting : FILE =  
/opt/NetApp/smo/mnt/_mnt_triggerfish_bt_bismarck_nfs_smoke_SMO_SMO  
KE_NFSTEST_BISMAR_NFSTEST_F_C_0_8abc01ec0e78f05c010e78f06f120001/d  
atafiles/users01.dbf
```

...

```
DBVERIFY - Verification complete  
Total Pages Examined          : 1280  
Total Pages Processed (Data)  : 0  
Total Pages Failing   (Data) : 0  
Total Pages Processed (Index): 0  
Total Pages Failing   (Index): 0  
Total Pages Processed (Other): 16  
Total Pages Processed (Seg)   : 0  
Total Pages Failing   (Seg)   : 0  
Total Pages Empty           : 1264  
Total Pages Marked Corrupt    : 0  
Total Pages Influx           : 0  
Highest block SCN             : 11133 (0.11133)
```

```
DBVERIFY - Verification starting : FILE =  
/opt/NetApp/smo/mnt/_mnt_triggerfish_bt_bismarck_nfs_smoke_SMO_SMO  
KE_NFSTEST_BISMAR_NFSTEST_F_C_0_8abc01ec0e78f05c010e78f06f120001/d  
atafiles/system01.dbf
```

...

```
DBVERIFY - Verification complete  
Total Pages Examined          : 110080  
Total Pages Processed (Data)  : 62640  
Total Pages Failing   (Data) : 0  
Total Pages Processed (Index): 11746  
Total Pages Failing   (Index): 0  
Total Pages Processed (Other): 1853  
Total Pages Processed (Seg)   : 0  
Total Pages Failing   (Seg)   : 0  
Total Pages Empty           : 33841  
Total Pages Marked Corrupt    : 0  
Total Pages Influx           : 0  
Highest block SCN             : 21591047 (0.21591047)  
Operation Id [8abc01ec0e78f2ae010e78f2c8710001] succeeded.
```

Restoring and recovering a database backup

Restoring and recovering backups

SnapManager provides a restore operation that includes a recover option. As a result, you can either use SnapManager to both restore and recover the backup, or you can use SnapManager to restore the backup and another tool, such as RMAN, to recover the data.

The SnapManager recover option is part of the restore operation, which means you must perform the restore and recover at the same time if you are using SnapManager. You cannot perform restore operation and then, at a later time perform a SnapManager recover operation.

Note

You can use any tool or script that you normally use to recover a database. You do not have to use SnapManager. You must, however, recover the database before you can use it.

The `smo backup restore` operation also gives you the option of deciding how much of the backup to restore. You can:

- ◆ Restore the entire backup.
- ◆ Restore only the tablespaces or data files you specify.
- ◆ Restore only the control files.

If you include the recover option on the command line, you can recover the database to:

- ◆ The last transaction that occurred in the database (all logs).
- ◆ A specific date and time.
- ◆ A specific Oracle System Change Number (SCN).
- ◆ The time of the backup (no logs).

Note

Both the date and time recovery and the SCN recovery are point-in-time recoveries.

For more information on how SnapManager handles recovering and restoring backup information, see “[Approaches to backing up, restoring, and recovering data](#)” on page 14.

General steps for restoring and recovering a backup

The basic steps for restoring a backup are as follows.

1. Specify the name of the backup. This can be either:
 - ❖ The label you supplied when you created the backup
 - ❖ The GUID that SnapManager generated when you created the backup. (For more information about the GUID, see [“General notes about the commands”](#) on page 134.)
2. Specify the name of the profile from which the backup was created.
3. Specify which components of the backup you want to restore:
 - ❖ The entire backup
 - ❖ Specific tablespace(s)
 - ❖ Specific data file(s)
 - ❖ Control files
4. (Optional) Specify that you want to recover the database:
 - ❖ The last transaction that occurred in the database (all logs).
 - ❖ A specific date and time.
 - ❖ A specific Oracle System Change Number (SCN).
 - ❖ The time of the backup (no logs).
5. (Optional) Decide if you want SnapManager to change the state of the database. If you include the `-force` flag, SnapManager changes the database state, if necessary. For example, when you include this option, if the database is offline and it needs to be online, SnapManager forces it online.

Restoring a backup procedure

To restore a backup, use the `smo backup restore` command. It has the following general format:

```
smo backup restore
  -profile profile_name {-label label_name | -id guid}
  {-complete | -files list | -tablespaces list | -controlfiles}
  [-recover {-nologs | -alllogs | -until {scn | date }}]
  [-using_backup_controlfile]
  [-force] [-quiet | -verbose]
```

For details on the options and arguments used in this command, see [“The smo backup restore and recovery command”](#) on page 148.

Example: This example restores the entire backup labeled `ProdFriday` that is part of the profile `Production`. It also recovers the database to the last system control number.

```
smo backup restore -profile Production -label ProdFriday -complete  
-recover -until scn
```


About this chapter

This chapter details how to use SnapManager to clone backups of database instances.

This chapter includes the following topics:

- ◆ [“General SnapManager database cloning information”](#) on page 78
- ◆ [“Creating a clone specification”](#) on page 80
- ◆ [“Cloning a database from a backup”](#) on page 88
- ◆ [“Cloning a database in its current state”](#) on page 90
- ◆ [“Options for cloning RAC databases”](#) on page 91

General SnapManager database cloning information

SnapManager's clone feature

SnapManager's clone feature lets you quickly create a copy of a database instance that is an exact replica of the original database. You create the clone from a full backup or from the current state of the database.

Using SnapManager to create the clone provides the following advantages:

- ◆ Speed. SnapManager can instantaneously clone data volumes.
- ◆ Space efficiency. Creating a clone using SnapManager requires essentially no overhead. The clone is a writable snapshot of the original database instance so, like all snapshots, it requires space only for the changes between the backup and the clone. It can also grow as needed. In contrast, a physical clone of the database requires that you have sufficient space available to copy the entire database.
- ◆ Virtual copy. The clone is a virtual copy of its original. For example, you can use a clone for such operations as testing and bug fixing, platform and update checks, multiple simulations against a large data set, and remote office testing and staging. People can work with the cloned database instance as if it were the original database; however, their changes to the clone do not affect the original.
- ◆ Simplicity. Use the `smo clone create` command.

Note

You must create a clone specification file before you execute the `smo clone create` command. For instructions on creating this file, see [“Creating a clone specification”](#) on page 80.

How the clone feature works

SnapManager creates a clone using a snapshot of the database. This differs from a traditional clone, which creates a physical copy of the entire original database instance. To create a physical copy, you must have space available that is equal to the current size of the database instance. For example, if your database instance is 17MB, you would need a minimum of an additional 17MB available to hold the copy. With SnapManager, you can create very space efficient clones very quickly. SnapManager's clone operation uses the FlexClone feature available with Data ONTAP (see the Data ONTAP documentation for more information on FlexClone).

The basic steps for creating a clone are as follows:

1. Create a full backup of the database instance while it is offline or online.
2. Create a clone specification file for the database instance. SnapManager creates the clone specification file from the information in the backup, or you can create the specification file. SnapManager uses this file to correctly create the clone, based on the information in the original database instance and your specifications.
3. Create the clone.

Key points about cloning

SnapManager supports creating a clone of a database from a backup. SnapManager also supports cloning from the current state of the database, which enables you to perform the backup and clone in one step.

SnapManager only supports cloning a database on the same host. You must give the clone a new Oracle System Identifier (SID). Oracle does not permit you to run two databases with the same SID simultaneously on the same host.

As part of the cloning process, SnapManager creates the necessary Oracle files and parameters for the cloned database. An example of a necessary Oracle file is:

```
init<clone_SID>.ora
```

When SnapManager clones the storage for a database, it creates a new file system mountpoint, but it does not change the directory structure under the mountpoint.

You can clone a Real Application Cluster (RAC) database instance as well as a non-clustered database instance. A RAC clone starts as a single database instance. For more information, see [“Options for cloning RAC databases”](#) on page 91.

Once the cloning operation completes, the new database is up and running.

Files created during the clone process

When you clone a database, SnapManager creates a new

```
init<clone_SID>.ora
```

file for the database in the \$ORACLE_HOME/dbs directory.

If the database you cloned was using an `spfile`, then SnapManager creates an `spfile` for the clone. It places this file in the \$ORACLE_HOME/dbs directory as well:

```
spfile<clone_SID>.ora
```

It also creates the `bdump`, `cdump`, and `udump` directories.

Creating a clone specification

About the clone specification

Before you can create a clone from the CLI, you must create a clone specification. You can choose any one of these:

- ◆ Use the `smo clone template` command to create a template, and then edit it.
- ◆ Identify an existing clone specification you have already created for SnapManager to use for the clone.

Note

This release of SnapManager uses clone specifications, which are not backwards compatible with clone maps from previous releases. You must recreate the clone map as a clone specification before using it to clone a backup.

- ◆ Create a new clone specification XML file. SnapManager uses this information to determine where to place the files it clones and how to handle dumps, control files, and parameters, and other information.
- ◆ Do not use spaces to create a location for the clone in the clone specification file.

Clone Specification Components

The clone specification is made up of two components:

- ◆ **Storage specification** - Specifies the locations for the new storage created for the clone, such as, data file mountpoints, ASM disk groups, and raw devices. These are the items that must be mapped from the source to the destination.

Note

You cannot specify the raw device mapping. SnapManager for Oracle or the operating system picks the next available device name and uses that for the cloned raw device.

- ◆ **Database specification** - Specifies the database options for the clone, such as, control files, redo logs, archive logs, and Oracle parameters.

Creating the clone specification

Create the clone specification file or edit the clone specification template with any text editor.

Use this command syntax to create the clone specification template:

```
sno clone template -profile profile_name -label label
```

The following is the clone specification file structure, showing the storage specification section and the database specification sections of the clone specification. Edit the template using this as a guide.

```
<clone-specification>
  <storage-specification/>
  <database-specification/>
</clone-specification>
```

Clone Specification Considerations:

- ◆ Use `<!--comment here--!>` to indicate comment lines (standard XML).
- ◆ (Recommended) Use an extension of `.xml` for the file to enable appropriate editing features.

Creating the Storage Specification Component:

The storage specification must include mappings for all storage components (file system mountpoints, ASM diskgroups, raw devices) that contain data files in the source database. There is no need to map storage components that just contain control files, archived logs, etc..

◆ Mountpoints

You must map the mountpoints for the data files from the source location in the backup to the destination location.

```
<mountpoint>
  <source>/mnt/weever/fearless_ES9</source>
  <destination>/mnt/weever/CLONE_ES9</destination>
</mountpoint>
```

◆ ASM disk group information

If you have data files in ASM disk groups in the source, map the name of the disk group from the source to the destination in the storage specification.

This is an example of the ASM disk group syntax to use in the clone specification:

```
<asm-diskgroup>
  <source>+DG1</source>
  <destination>+CLONE_DG1</destination>
</asm-diskgroup>
```

◆ Raw Device mapping information

Unlike in the clone mapping file from previous versions of SnapManager for Oracle, you cannot specify a location for the raw device on the destination. This is because some platforms do not allow users to assign raw device labels. SnapManager for Oracle chooses the next available device name for the cloned raw device. If you have a raw device on the source, you must specify the path for the raw device on the source, and then specify `destination auto-generate="true"` for the destination.

This is an example of the raw device syntax to use in the clone specification.

```
<raw-device>
  <source>/dev/raw/raw1</source>
  <destination auto-generate="true"/>
</raw-device>
```

Example of a completed storage specification

This is an example of the storage specification to include in the clone specification.

```
<storage-specification>
  <storage-mapping>
    <mountpoint>
      <source>/mnt/weever/fearless_ES9</source>
      <destination>/mnt/weever/CLONE_ES9</destination>
    </mountpoint>
    <asm-diskgroup>
      <source>+DG1</source>
      <destination>+CLONE_DG1</destination>
    </asm-diskgroup>
    <raw-device>
      <source>/dev/raw/raw1</source>
      <destination auto-generate="true" />
    </raw-device>
    <raw-device>
      <source>/dev/raw/raw2</source>
      <destination auto-generate="true" />
    </raw-device>
  </storage-mapping>
</storage-specification>
```

Creating the Database Specification Component:

The database specification includes:

◆ Control file information

As part of the database specification to be included in the clone specification, identify a list of the control files to create for the clone.

```
<controlfiles>
  <file>/mnt/weever/CLONE_ES9/control/control01.ctl</file>
  <file>/mnt/weever/CLONE_ES9/control/control02.ctl</file>
</controlfiles>
```

◆ Redo log information

Specify the redo log structure for the clone.

```
<redologs>
  <redogroup>
    <file>/mnt/weever/CLONE_ES9/redo/redo01.log</file>
    <number>1</number>
    <size unit="MB">100</size>
  </redogroup>
  <redogroup>
    <file>/mnt/weever/CLONE_ES9/redo/redo02.log</file>
    <number>2</number>
    <size unit="MB">100</size>
  </redogroup>
</redologs>
```

◆ Oracle parameters

The parameters you must specify for the destination are:

- ❖ Background dump, core dump, and user dump

Optional parameters to specify for the destination are:

- ❖ Archive logs

If you do not specify archive logs for the destination, SnapManager for Oracle creates the clone in NOARCHIVELOG mode.

- ❖ Any valid `init.ora` parameter

SnapManager copies this parameter information into the clone's `init.ora` file.

Note

If a parameter entered conflicts with a parameter that SnapManager for Oracle must change to create the clone, the clone operation is aborted and you receive an error message.

```

<parameters>
  <parameter>
    <name>log_archive_dest_1</name>
    <value>LOCATION=/mnt/weever/CLONE_ES9/archive</value>
  </parameter>
</parameters>

```

Example of a completed database specification

The following is an example of a database specification structure, including control files, redo logs, and parameters.

```

<database-specification>
  <controlfiles>
    <file>/mnt/weever/CLONE_ES9/control/control01.ctl</file>
    <file>/mnt/weever/CLONE_ES9/control/control02.ctl</file>
  </controlfiles>
  <redologs>
    <redogroup>
      <file>/mnt/weever/CLONE_ES9/redo/redo01.log</file>
      <number>1</number>
      <size unit="MB">100</size>
    </redogroup>
    <redogroup>
      <file>/mnt/weever/CLONE_ES9/redo/redo02.log</file>
      <number>2</number>
      <size unit="MB">100</size>
    </redogroup>
  </redologs>

```

Table continued on next page...

(Table continued)

```
<parameters>
  <parameter>
    <name>log_archive_dest_1</name>
    <value>LOCATION=/mnt/weever/CLONE_ES9/archive</value>
  </parameter>
  <parameter>
    <name>background_dump_dest</name>
    <value>/mnt/weever/CLONE_ES9/admin/bdump</value>
  </parameter>
  <parameter>
    <name>core_dump_dest</name>
    <value>/mnt/weever/CLONE_ES9/admin/cdump</value>
  </parameter>
  <parameter>
    <name>user_dump_dest</name>
    <value>/mnt/weever/CLONE_ES9/admin/udump</value>
  </parameter>
</parameters>
</database-specification>
```

Creating Arbitrary SQL Statements (Optional):

In addition to parameters, you can specify arbitrary SQL statements to execute against the clone once it is online. One example of a use for this is to fix issues like recreating the temp files in the cloned database.

The following is an example of what you would enter after the other parameters in the database specification portion of the clone specification file.

```
<sql-statements>
  <sql-statement>
    ALTER TABLESPACE TEMP ADD
    TEMPFILE '/mnt/weeever/CLONE_ES9/temp_eric01.dbf'
    SIZE 41943040 REUSE AUTOEXTEND ON NEXT 655360
    MAXSIZE 32767M
  </sql-statement>
</sql-statements>
```

Clone a backup of a database instance using the clone specification. Refer to [“Cloning a database from a backup”](#) on page 88 or [“Cloning a database in its current state”](#) on page 90 for information about creating clones.

Cloning a database from a backup

General steps for cloning a database from a backup

You have two options when cloning.

- ◆ Clone the database from its current state, which performs the backup and cloning with one command. Refer to [“Cloning a database in its current state”](#) on page 90 for more information.
- ◆ Clone from an existing backup.

This section takes you through the basic steps to clone a database from a backup. For a syntax example, see [“Cloning a database from a backup procedure”](#) on page 89

The basic steps for cloning a database instance you have backed up are as follows.

1. Specify the name of the backup. This can be either:
 - ❖ The label you supplied when you created the backup
 - ❖ The 32-character HEX string GUID that SnapManager generated when you created the backup. (For more information about the GUID, see [“General notes about the commands”](#) on page 134.)
2. Supply a SID for the new, cloned database.
 - ❖ If you are cloning a RAC database and you want the clone to be a RAC database, enter the database name of the new RAC database.
 - ❖ If you want the clone to be a non-RAC database, enter the SID for the database.

Note

Oracle does not let you run two databases with the same SID simultaneously on the same host. Because of this, you must supply a new SID for each clone.

3. Specify the name of the profile from which the backup was created.
4. Specify the path to the file containing the clone specification information. This can be an absolute or relative path. You must set up the clone specification file, or have SnapManager create one, before you start the clone operation.

When the cloning operation completes, the new database instance is up and running.

Note

After performing a clone, put the new database SID in the Oracle `oratab` file. SnapManager only performs backup operations on databases that are listed in the `oratab` file. You should also add the database to your `TNSNAMES` file on your client machines.

Cloning a database from a backup procedure

To use the CLI `smo clone create` command to clone a backup of a database instance, complete the following steps.

Note

Make sure that the backup you are cloning is a full backup.

Step	Action
1	Create a clone specification file. For instructions on how to do this, see “Creating a clone specification” on page 80.
2	Enter the <code>smo clone create</code> command. It has the following general format: <pre>smo clone create {-id <i>guid</i> -label <i>label_name</i>} -newsid <i>new_sid</i> -profile <i>profile_name</i> -clonespec <i>full_path_to_clonespecfile</i> [-quiet -verbose]</pre> For details on the options and arguments used in this command, see “The <code>smo clone create</code> command” on page 155. Example: The following example creates a clone from a backup labeled <code>Test1</code>, which is included in the profile <code>alltests</code>. The clone uses the clone specification file. <pre>smo clone create -label Test1 -newsid EXC79D -profile alltests -clonespec ./maps/clonespec.xml</pre>

Cloning a database in its current state

Cloning a database in its current state

In this SnapManager for Oracle release, you can create a backup and clone in one procedure from current state of the database. Instead of specifying a backup to be used for the clone using either the `-id` or `-label` flags, you specify the profile with the flag `-current`, which tells SnapManager for Oracle to clone from the current state of the database (creating a backup and then the clone with one command).

Cloning a database in its current state procedure

To clone the database from its current state, use this command syntax:

```
smo clone create -profile profile_name -current -newsid clone_name
-clonespec ./clonespec_filename.xml
```

This command takes a backup (autogenerating the label for the backup) and immediately makes a clone from that backup, using an existing clone specification which you identify.

Cloning with RAC databases

Options for cloning RAC databases

SnapManager clones a RAC database to a non-RAC database and sets the Oracle parameter `cluster.database` to `false`. To use the cloned database as a non-RAC database, follow the steps in “[Cloning a RAC database to a non-RAC database](#)” on page 91.

To use the cloned database as a RAC database, follow the steps in “[Cloning a RAC database and changing it to a RAC database procedure](#)” on page 91.

Cloning a RAC database to a non-RAC database

Follow this procedure to clone the database as a non-RAC database. To start and stop the database, set `ORACLE_SID` parameter to the SID of the cloned database and run `SQLPlus`. For example, you could enter the following command lines:

```
$ smo clone create -label rac_full_backup -profile rac_profile
-newsid clone1 -clonespec rac_nfs.xml -verbose

$ export ORACLE_SID=clone1

$ sqlplus

...
```

Cloning a RAC database and changing it to a RAC database procedure

Follow this procedure to clone a RAC database and then start a one node RAC instance, complete the following steps.

Step	Action
1	Use the database name of the new RAC database instance as the value for the <code>-newsid</code> option in the SnapManager <code>smo clone create</code> command. Note Do not use the Oracle SID as the name of the new RAC database instance.

Step	Action
2	When the cloning operation completes, rename the file from: <pre>init<clone_SID>.ora</pre> to: <pre>init<clone_local_instance_SID>.ora</pre> Edit that file to set the RAC database instance information.
3	Register the cloned RAC database with <code>srvctl</code> .

Example of setting up a cloned database

The following command lines set up a cloned database as a one node RAC instance. You need to use an editor, such as `vi`, to change the `.ora` file.

```
$ smo clone create -label rac_full_backup -profile rac_profile
  -newsid CLONE1 -clonespec rac_nfs.clonespec
  -verbose
$ export ORACLE_SID=CLONE1
$ sqlplus
      shutdown immediate;
$ vi initCLONE11.ora
      *.cluster_database = TRUE
      CLONE11.instance_number=1
      CLONE12.instance_number=2
      CLONE11.thread=1
      CLONE12.thread=2
      CLONE11.undo_tablespace='UNDOTBS1'
      CLONE12.undo_tablespace='UNDOTBS2'
$ srvctl add database -d CLONE1 -o
/u02/app/oracle/product/10.2.0/db -m rtp.netapp.com -r primary -y
manual
$ srvctl add instance -d CLONE1 -i CLONE11 -n anzio
$ srvctl start instance -d CLONE1 -i CLONE11
```

About this chapter

This chapter covers the tasks you can perform to help you manage SnapManager operations.

This chapter includes the following topics:

- ◆ [“Aborting an operation”](#) on page 94
- ◆ [“Deleting repositories”](#) on page 97
- ◆ [“Changing an existing profile”](#) on page 98
- ◆ [“Deleting a profile”](#) on page 99
- ◆ [“Displaying information about backups”](#) on page 100
- ◆ [“Deleting backups”](#) on page 101
- ◆ [“Mounting and unmounting backups”](#) on page 103
- ◆ [“Checking the version of SnapManager”](#) on page 106
- ◆ [“Uninstalling SnapManager software”](#) on page 107

Aborting an operation

SnapManager's abort feature

SnapManager lets you abort any of its operations that are currently executing from the CLI as well as any operation that was in progress when an interrupt, such as a system crash, occurred.

The abort operation lets SnapManager perform a graceful cleanup of the operation. This cleanup is necessary because SnapManager maintains information on current operations persistently in order to prevent two incompatible operations from executing simultaneously. For example, SnapManager does not let you perform a restore operation if a backup operation is in progress for that database. If the backup operation was incomplete, you need to perform a abort operation to release the resources associated with that operation.

When you perform an abort operation, SnapManager updates its persistent state information. Without the cleanup, you would not be able to perform additional operations.

Using the abort command

You can use the `smo operation abort` command to:

- ◆ Clean up after an incomplete backup

SnapManager automatically attempts to abort a backup operation if it fails due to an external cause, such as losing the connection to the storage system. If this happens, there may still be resources being used by the backup operation. To clean up these resources, you need to run the abort operation against the failed backup. For example, the abort operation would unlock the target database for other operations and update the repository metadata regarding the failed operation.

For more information SnapManager's states, such as an incomplete backup, see "[SnapManager operation states](#)" on page 25.

- ◆ **(Not recommended)** Abort a backup while it is running

When you abort a backup, SnapManager attempts the following:

- ❖ To kill the SnapManager processes
- ❖ To cleanly abort any pending SnapManager backup operations on the target database and/or on the storage systems.
- ❖ To unlock the target database for other SnapManager operations. For example, suppose that the backup operation put a tablespace in backup

mode. The `smo operation abort` command takes the tablespace out of backup mode. If a snapshot was created during the backup operation, the `smo operation abort` command removes that snapshot from the storage system.

There are two important cautions to keep in mind if you abort a backup that is in progress;

- ◆ If you run the abort operation against a backup that is performing a verification (i.e., the `verify` option is enabled), SnapManager aborts both the verification and the backup.
- ◆ If the abort operation fails, you may need to perform a manual cleanup if SnapManager fails to abort all actions of the operation. For example, you may need to take a tablespace out of backup mode, remove a useless snapshot, or remove the target database lock file.

General steps for aborting an operation

This section takes you through the basic steps to abort an operation. For a syntax example, see “[Aborting an operation](#)” on page 95.

The basic steps for aborting an operation are as follows:

1. Specify the label or GUID for the operation you want to abort. This can be either:
 - ❖ If you are aborting a backup, specify the label you supplied when you created the backup
 - ❖ If you are aborting another operation, specify the GUID for the operation. (For more information about the GUID, see “[General notes about the commands](#)” on page 134.)
2. Specify the name of the profile associated with the operation.
3. (Optional) Specify whether you want SnapManager to force the abort to happen. If you force the abort, you may need to perform a manual cleanup if SnapManager fails to abort all actions of the operation.
4. (Optional) Specify how you want SnapManager to handle messages when executing the command from the CLI. For more information on these options, see “[Common options for handling messages shown on the console](#)” on page 136.

Aborting an operation

To abort a backup from the CLI, use the `smo operation abort` command. It has the following general format:

```
smo operation abort {-id backup-id |-label label_name}  
                    -profile profile_name [-force] [-quiet | -verbose]
```

For details on the options and arguments used in this command, see “[The smo operation abort command](#)” on page 161.

Deleting repositories

About deleting a repository

Deleting a repository removes all of the SnapManager metadata in the repository, as well as the repository schema itself. Manually remove the actual resources represented by the metadata (profiles, backups, clones) before deleting a repository.

Attention

This should be done with caution because once the repository is deleted, you cannot recover it or any of the associated profiles, backups and clones.

Deleting a repository procedure

To delete the repository, use the `smo repository delete` command.

```
# smo repository delete
  {-repository -dbname db_name -login -username username -password
password -port port_number -host host}
  [-sid sid]
  [-noprompt]
  [-quiet | -verbose]
```

Example: The following command line deletes the repository created in the previous example.

```
# smo repository delete -repository -dbname DB1 -login -username
user1 -password high -host host1 -port 1521
```

For more information about this command, see [“The smo repository delete command”](#) on page 184.

Changing an existing profile

Once you have created a profile, you can modify it. For example, if the Oracle database password information changes, you must change this information in the profile.

To modify the profile from the CLI, you use the `smo profile update` command. It has the following general format:

```
smo profile update
-profile profile_name
{-repository -dbname db_name -host repo_host -login
-password repo_password -username repo_username -port repo_port}
{-database -dbname dbname -host dbhost -login -password db_password
-username db_username -port db_port [-sid db_sid]}
{-osaccount account_name}
[-rman {-tnsname tnsname -login -password rman_password -username
rman_username} | -controlfile ]
[-asm -host asm_host -login -password asm_password -username
asm_username -port asm_port -sid asm_sid ]
[-retain number]
[-quiet | -verbose]
```

For details on the options and arguments used in this command, see “[The smo profile update command](#)” on page 176.

Examples: The following command line updates the database information for the profile MYTEST. It supplies a new username/password pair, a new host, and a new port number.

```
# smo profile update -profile MYTEST -repository -dbname database1
-login -username sys -password pacific -host ocean -port 1521
```

Deleting a profile

You can delete the profile at any time. To delete the profile, use the `smo profile delete` command. It has the following general format:

```
smo profile delete
-profile profile_name
[-quiet] [-verbose]
```

For details on the options and arguments used in this command, see “[The smo profile delete command](#)” on page 170

Example: The following command line deletes the profile MYTEST.

```
# smo profile delete -profile MYTEST
```

Displaying information about backups

Listing information in current backups

You can quickly check on which backups were made for a profile and what their state is by executing the `smo backup list` command. For each profile, the `smo backup list` command displays the information on the most recent backup first and then continues until it has displayed information on all the backups. It ends with the oldest backup. This operation displays the following information for each backup:

- ◆ The database name, host name, and profile name
- ◆ The date and time the operation was performed
- ◆ For each operation, it displays:
 - ❖ The backup ID
 - ❖ The label
 - ❖ Any comment
 - ❖ Whether the operation succeeded or failed
 - ❖ A list of the storage system/volume/snapshots created
 - ❖ A list of the mountpoints comprising the backup
 - ❖ Information about whether the backup was verified
 - ❖ Information about whether the backup was used to create a database clone

General steps for listing backup information

This section takes you through the basic steps to list backup information. For a syntax example, see “[Listing backup information procedure](#)” on page 100.

The basic steps for listing information about backups are as follows.

1. Specify the name of the profile associated with the backup you want to view.

Listing backup information procedure

To display information about backups in a particular profile, use the `smo backup list` command. It has the following general format:

```
smo backup list -profile profile_name
[-quiet | -verbose]
```

For details on the options and arguments used in this command, see “[The smo backup list command](#)” on page 144.

Deleting backups

Using the delete backups feature

You can delete backups for a database at any time. It's a good practice to delete a backup when you no longer need it to free up the space it occupied. There is a limit on the number of snapshots per volume that you can have. If you remove backups you no longer need, you reduce the chance of reaching the limit of 255 snapshots.

SnapManager's backup delete operation lets you specify which backups you want to remove for a specific profile. You can use this operation even if you specified a number of successful backups for SnapManager to retain when you set up the profile.

Note

SnapManager uses a sliding window approach to retaining backups. It keeps successful backups based on the value you supply for the profile operation's retain option. It also keeps a list of failed backups. The failed backups do not affect the space on the disk. For more information on how SnapManager handles backups, see [“SnapManager's policy for retaining backups”](#) on page 23.

General steps for deleting a backup

This section takes you through the basic steps. For a syntax example, see [“Deleting a backup procedure”](#) on page 102.

The basic steps for removing a backup are as follows:

1. Specify either the name of the backup you want to delete or the number of backups you want to keep.
The name of the backup can be either:
 - ❖ The label you supplied when you created the backup
 - ❖ The GUID.
2. Specify whether you want SnapManager to retain any of the backups. If you set the `-retain` option to 0, SnapManager removes all the backups.
3. Specify the name of the profile from which the backup was created.
4. (Optional) Specify whether you want SnapManager to force the removal of the backup.

Deleting a backup procedure

To remove a backup, use the `smo backup delete` command. It has the following general format:

```
smo backup delete {-label label_name | -id backup_guid } |-all  
  {-retain number} -profile profile_name [-force]  
  {[-quiet | -verbose]}
```

For details on the options and arguments used in this command, see “[The smo backup delete command](#)” on page 142.

Mounting and unmounting backups

Using the mount and unmount features

When you use SnapManager, it automatically handles mounting a backup to make it available to the host or unmounting a backup to make it unavailable.

SnapManager provides the mount and unmount operations for situations where you are using an external tool such as RMAN to access the files in the backup. In that case, you can use the mount operation to change the state of a backup to allow access and the unmount operation to change the state of a backup to deny access.

The `smo backup mount` command displays a list of paths where the snapshots comprising the backup have been mounted.

The `smo backup unmount` command lets you free up resources on the host when the backup is no longer needed for external operations.

General steps for mounting and unmounting backups

This section takes you through the basic steps. For a syntax example, see [“Mounting and unmounting backups procedure”](#) on page 104.

Whether you are performing a mount or unmount operation, the basic steps are the same. You need to:

1. Specify the name of the backup you are working with. This can be either:
 - ❖ The label you supplied when you created the backup
 - ❖ The GUID that SnapManager generated when you created the backup. (For more information about the GUID, see [“General notes about the commands”](#) on page 134.)
2. Specify the name of the profile from which the backup was created.
3. (Optional) Specify whether you want SnapManager to force the operation (either mount the backup or unmount it).
4. (Optional) Specify how you want SnapManager to handle messages when executing the command from the CLI. For more information on these options, see [“Common options for handling messages shown on the console”](#) on page 136.

Mounting and unmounting backups procedure

To mount a backup from the CLI, use the `smo backup mount` command. It has the following general format:

```
smo backup mount {-id backup_id |-label label_name}  
                  -profile profile_name [-quiet | -verbose]
```

For details on the options and arguments used in this command, see “[The smo backup mount command](#)” on page 146.

To unmount a backup from the CLI, use the `smo backup unmount` command. It has the following general format:

```
smo backup unmount {-id backup_id |-label label_name}  
                   -profile profile_name [-force] [-quiet | -verbose]
```

If you use the `-force` option, SnapManager for Oracle tries to unmount the backup and clean up any associated resources. The log shows the unmount operation as successful, but you may have to manually clean up resources if there were errors in the log.

For details on the options and arguments used in this command, see “[The smo backup unmount command](#)” on page 151.

Examples:

The following is an example of a mount operation.

```
# smo backup mount -label test -profile gal -verbose
```

Output:

```
SMO-13039 [INFO ]: Operation GUID: 8abc01b30844958e01084495a3b40001  
SMO-08024 [INFO ]: Beginning to connect snapshot  
SMO_ESDB1_P_C_20051219142901EST_8abc01b308447d850108447dc8bb0002  
of mount(s) [/u02/app/oracle/esdb1].  
SMO-08026 [INFO ]: Done connecting snapshot  
SMO_ESDB1_P_C_20051219142901EST_8abc01b308447d850108447dc8bb0002  
of mount(s) [/u02/app/oracle/esdb1].  
SMO-13011 [INFO ]: Successfully completed operation: Manage  
SMO-13036 [INFO ]: Operation Status: SUCCESS  
SMO-13037 [INFO ]: Elapsed Time: 0:00:31.086
```

The following is an example of an unmount operation:

```
# smo backup unmount -label test -profile gal -verbose
```

Output:

```
SMO-13039 [INFO ]: Operation GUID: 8abc01b308449e7d0108449e94f80001
```

```
SMO-08027 [INFO ]: Beginning to disconnect connected mount(s)
[/u02/app/oracle/esdb1_SMO_ESDB1_P_C_20051219142901EST_8abc01b3084
47d850108447dc8bb0002].
SMO-08029 [INFO ]: Done disconnecting connected mount(s)
[/u02/app/oracle/esdb1_SMO_ESDB1_P_C_20051219142901EST_8abc01b3084
47d850108447dc8bb0002].
SMO-13011 [INFO ]: Successfully completed operation: Manage
SMO-13036 [INFO ]: Operation Status: SUCCESS
SMO-13037 [INFO ]: Elapsed Time: 0:00:32.435
```

Checking the version of SnapManager

Determining the version of SnapManager

SnapManager provides a version operation that lets you determine which version of the product you are running on your local host.

To determine which version of SnapManager the host is running, execute the `smo version` command. It has the following format:

```
smo version [-quiet | -verbose]
```

Uninstalling SnapManager software

Uninstalling the software from the host

Complete the following steps to uninstall the SnapManager software from your Linux/UNIX host.

Step	Action
1	Log in as root.
2	Use the following command to stop the SnapManager for Oracle Server. <pre># smo_server stop</pre>
3	Use the following command to remove the SnapManager software. <pre># UninstallSmo</pre>
4	After the introduction text, press Enter to continue. The uninstallation completes.

Uninstalling the GUI software on a Windows system

If the GUI is currently running, exit from the GUI before uninstalling the GUI software.

There are two ways to uninstall the SnapManager software from a Windows machine:

- ◆ Go to **Start > Programs > NetApp > SnapManager for Oracle GUI 2.0> Uninstall SMO**.
- ◆ Go to **Start > Control Panel**. Double click **Add or Remove Programs** and select **NetApp SnapManager for Oracle GUI 2.0**, and then click **Change/Remove**.

About this chapter

This chapter covers one of the more advanced procedures you can perform using SnapManager.

This chapter includes the following topic:

- ◆ [“Performing a block level restore using RMAN”](#) on page 110

Performing a block level restore using RMAN

Performing a block level restore using RMAN tasks

The following are the basic tasks involved in backing up a database using SnapManager for Oracle and recovering that backup using RMAN:

1. Perform a full offline backup of the database using SnapManager for Oracle.
2. Verify the backup is catalogued with RMAN.
3. Verify the blocks have been backed up correctly.
4. Mount the backup using SnapManager for Oracle.
5. Using RMAN, perform the block restore.
6. Verify the blocks have been repaired.

Task 1: Performing a full offline backup using SnapManager for Oracle

To perform the backup using SnapManager for Oracle, use this command syntax:

```
smo backup create -offline -full -profile profilename -label  
backupname -verbose
```

Example:

```
smo backup create -offline -full -profile e -label full_backup  
-verbose
```

Output:

```
SMO-07109 [INFO ]: Cataloguing all files in backup set with RMAN,  
TAG=SMO_full_backup_1158773581857, RMAN=ES0/controlfile.  
...  
SMO-13037 [INFO ]: Successfully completed operation: Backup  
SMO-13048 [INFO ]: Operation Status: SUCCESS  
SMO-13049 [INFO ]: Elapsed Time: 0:02:20.506  
Operation Id [ff8080810dcc47e3010dcc47eb7a0001] succeeded.
```

Task 2: Verifying the backup is catalogued with RMAN

To verify the backup is catalogued with RMAN, from the database host, type this command:

```
rman target /
```

Output:

```

Recovery Manager: Release 10.2.0.1.0 - Production on Wed Sep 20
10:33:41 2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.
connected to target database: ES0 (DBID=2972033709, not open)
RMAN> list datafilecopy tag SMO_full_backup_1158773581857;
using target database control file instead of recovery catalog
List of Datafile Copies
Key       File S Completion Time Ckp SCN       Ckp Time       Name
-----
335       1      A 20-SEP-06          1347825      20-SEP-06
/opt/NetApp/smo/mnt/_mnt_weever_revenge_ES0_SMO_E_ES0_F_C_0_ff8080
810dcc47e3010dcc47eb7a0001/system01.dbf
336       2      A 20-SEP-06          1347825      20-SEP-06
/opt/NetApp/smo/mnt/_mnt_weever_revenge_ES0_SMO_E_ES0_F_C_0_ff8080
810dcc47e3010dcc47eb7a0001/undotbs01.dbf
334       3      A 20-SEP-06          1347825      20-SEP-06
/opt/NetApp/smo/mnt/_mnt_weever_revenge_ES0_SMO_E_ES0_F_C_0_ff8080
810dcc47e3010dcc47eb7a0001/sysaux01.dbf
333       4      A 20-SEP-06          1347825      20-SEP-06
/opt/NetApp/smo/mnt/_mnt_weever_revenge_ES0_SMO_E_ES0_F_C_0_ff8080
810dcc47e3010dcc47eb7a0001/user01.dbf
337       5      A 20-SEP-06          1347825      20-SEP-06

RMAN>

```

Task 3: Verifying database backup block condition

To verify the database and determine if any blocks are corrupted, use this command syntax:

Example:

```
dbv FILE=user01.dbf
```

Output:

```

DBVERIFY: Release 10.2.0.1.0 - Production on Wed Sep 20 13:35:44
2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.

DBVERIFY - Verification starting : FILE = user01.dbf
Page 625 is marked corrupt
Corrupt block relative dba: 0x01400271 (file 5, block 625)
Bad header found during dbv:
Data in bad block:
  type: 240 format: 6 rdba: 0xed323b81
  last change scn: 0x6f07.faa74628 seq: 0x87 flg: 0x02
  spare1: 0x60 spare2: 0x5 spare3: 0xef7d

```

consistency value in tail: 0xa210fe71
check value in block header: 0x13c7
block checksum disabled

Page 626 is marked corrupt

Corrupt block relative dba: 0x01400272 (file 5, block 626)

Bad header found during dbv:

Data in bad block:

type: 169 format: 2 rdba: 0x5b886118
last change scn: 0x9bfb.a2ffb25c seq: 0xd1 flg: 0xb4
spare1: 0x38 spare2: 0xed spare3: 0xf9cb
consistency value in tail: 0xfc3469f2
check value in block header: 0xf570
computed block checksum: 0x98be

Page 627 is marked corrupt

Corrupt block relative dba: 0x01400273 (file 5, block 627)

Bad header found during dbv:

Data in bad block:

type: 158 format: 7 rdba: 0x2101e16d
last change scn: 0xe828.42414628 seq: 0xb4 flg: 0xff
spare1: 0xcc spare2: 0x81 spare3: 0x8665
consistency value in tail: 0x46d20601
check value in block header: 0x1a84
computed block checksum: 0x6c30

DBVERIFY - Verification complete

Total Pages Examined : 1280
Total Pages Processed (Data) : 1123
Total Pages Failing (Data) : 0
Total Pages Processed (Index): 0
Total Pages Failing (Index): 0
Total Pages Processed (Other): 34
Total Pages Processed (Seg) : 0
Total Pages Failing (Seg) : 0
Total Pages Empty : 120

Total Pages Marked Corrupt: 3

Total Pages Influx : 0
Highest block SCN : 1337349 (0.1337349)
[oracle@revenge revenge_ES0]\$

Note

Note that 3 pages are marked as corrupt.

Task 4: Mounting the backup using SnapManager for Oracle

To make the files from the backup accessible on the host and to RMAN, mount the backup using this command:

```
smo backup mount -profile profilename -label backupname -verbose
```

Example:

```
smo backup mount -profile e -label full_backup -verbose
```

Output:

```
SMO-13046 [INFO ]: Operation GUID: ff8080810dcc4ce6010dcc4cf0f00001
SMO-08052 [INFO ]: Beginning to connect mount(s)
[/mnt/weever/revenge_ES0] from logical snapshot
SMO_E_ES0_F_C_0_ff8080810dcc47e3010dcc47eb7a0001.
SMO-08025 [INFO ]: Beginning to connect mount
/mnt/weever/revenge_ES0 from snapshot
SMO_E_ES0_F_C_0_ff8080810dcc47e3010dcc47eb7a0001_0 of volume
revenge_ES0.
SMO-08027 [INFO ]: Finished connecting mount
/mnt/weever/revenge_ES0 from snapshot
SMO_E_ES0_F_C_0_ff8080810dcc47e3010dcc47eb7a0001_0 of volume
revenge_ES0.
SMO-08053 [INFO ]: Finished connecting mount(s)
[/mnt/weever/revenge_ES0] from logical snapshot
SMO_E_ES0_F_C_0_ff8080810dcc47e3010dcc47eb7a0001.
SMO-13037 [INFO ]: Successfully completed operation: Backup Mount
SMO-13048 [INFO ]: Operation Status: SUCCESS
SMO-13049 [INFO ]: Elapsed Time: 0:00:14.258
Operation Id [ff8080810dcc4ce6010dcc4cf0f00001] succeeded.
```

Task 5: Performing the block restore using RMAN

To recover the blocks, in RMAN, use this command syntax:

```
blockrecover datafile '/mountpoint/path/file.dbf' block block_id,
from tag backup_rman_tag
```

Example:

```
RMAN> blockrecover datafile '/mnt/weever/revenge_ES0/file01.dbf'
block 625, 626, 627 from tag SMO_full_backup_1158773581857;
```

Output:

```
Starting blockrecover at 20-SEP-06
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=153 devtype=DISK
channel ORA_DISK_1: restoring block(s) from datafile copy
/opt/NetApp/smo/mnt/_mnt_weever_revenge_ES0_SMO_E_ES0_F_C_0_ff8080
810dcc47e3010dcc47eb7a0001/user01.dbf
```

```
starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished blockrecover at 20-SEP-06
```

Task 6: Verifying the blocks have been repaired

To verify the blocks have been repaired, use the following command syntax:

```
dbv FILE=filename.dbf
```

Example:

```
dbv FILE=user01.dbf
```

Output:

```
DBVERIFY: Release 10.2.0.1.0 - Production on Wed Sep 20 13:40:01
2006
Copyright (c) 1982, 2005, Oracle. All rights reserved.
DBVERIFY - Verification starting : FILE = user01.dbf

DBVERIFY - Verification complete

Total Pages Examined          : 1280
Total Pages Processed (Data)  : 1126
Total Pages Failing   (Data)  : 0
Total Pages Processed (Index): 0
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 34
Total Pages Processed (Seg)   : 0
Total Pages Failing   (Seg)   : 0
Total Pages Empty           : 120
Total Pages Marked Corrupt    : 0
Total Pages Influx          : 0
Highest block SCN            : 1337349 (0.1337349)
```

Note

Note that no pages are corrupt.

All corrupted blocks have been repaired and restored.

About this chapter

This chapter includes information related to troubleshooting SnapManager for Oracle.

The chapter includes the following topics:

- ◆ [“Questions to consider”](#) on page 116
- ◆ [“Gathering troubleshooting information”](#) on page 118
- ◆ [“General troubleshooting tips for using the GUI”](#) on page 121
- ◆ [“General troubleshooting tips for working with clones”](#) on page 122
- ◆ [“General troubleshooting tips for SnapManager”](#) on page 124
- ◆ [“Common error conditions”](#) on page 129
- ◆ [“Where to go for more information”](#) on page 131

Questions to consider

Things to check

If a problem occurs when using SnapManager, the following list of questions could help you resolve the issue:

- ◆ Are the target database and the listener running?

To check this, run the command:

```
lsnrctl status.
```

You want to make sure that the database instance is registered with the listener.

- ◆ Is the storage visible and writable?

Use the commands `df` and `sanlun lun show` to see if the storage is visible. If so, go to the directory and click on one of the files in the directory to make sure that you can write to the storage system.

- ◆ Is the smo server running?

To check the status of the server, run the command:

```
smo_server status
```

To stop the smo server, run the command:

```
smo_server stop
```

To start the smo server, run the command:

```
smo_server start
```

- ◆ Can you run the `smo system verify` command?

This command checks to make sure that all the components of the environment required to run SnapManager are set up correctly. It has the format:

```
smo system verify [-quiet | -verbose]
```

For more information on this command, see “[The smo system verify command](#)” on page 189.

- ◆ Do you have the correct version of SnapManager?

Use the `smo version` command to check your version of SnapManager.

- ◆ Is there information in the SnapManager log files that is useful?

SnapManager records all log entries and places them into one set of rotating log files. The log files are found by default in:

- ❖ `/opt/NTAPsmo/log` on a Solaris host
- ❖ `/opt/NetApp/smo/log` on all other UNIX hosts

Each operation goes into its own log file of the form

`opFactory_server_id.log`.

- ❖ The server is the name of the server where the smo server is running.
- ❖ The id is a unique identifier that makes the log file unique.
- ◆ If you have archive logs stored on a storage system that is not running Data ONTAP, you should exclude them from consideration for backup with SnapManager for Oracle. The `smo.config` file enables you to exclude certain archive log files. The file is located in:
`/opt/NetApp/smo/properties/smo.config`
Follow the format suggested within the file to exclude the local archive logs.

Gathering troubleshooting information

Getting information using the dump command

SnapManager can create a compressed log file containing information about the SnapManager and its environment. This command captures information about any operation performed. SnapManager includes extra diagnostic information in the file. The information in the file includes:

- ◆ The steps performed
- ◆ How long they took
- ◆ What the outcome of each step was
- ◆ Any messages or warnings that occurred during the operation

Even if the SnapManager for Oracle server is not running, the dump information is still available by using the CLI command or from the GUI by using the right click menu and selecting **Create Diagnostics**.

If you encounter a problem with SnapManager that you cannot resolve, you can send this file to NetApp Global Services for help.

Locating the dump file

The dump results file contains the output for the dump request. The name of the file depends on the information supplied.

- ◆ If you did not specify an operation ID (the GUID or a label), the file is named `smo_dump_results.jar`

Information included in the dump file

SnapManager gathers information about the operation and your environment and places it in the dump results file. You can use the `jar` command to extract information from the dump file.

The following information is also included in the file:

- ◆ Operating system version
- ◆ Environment variables
- ◆ Java version
- ◆ SnapManager version
- ◆ SnapManager preferences
- ◆ SnapManager messages
- ◆ log4j properties

- ◆ SnapDrive version
- ◆ Oracle version
- ◆ Storage system version for the storage system
- ◆ Oracle oratab file
- ◆ Oracle listener status
- ◆ Oracle network configuration files (`listener.ora` and `tnsnames.ora`)
- ◆ Profile descriptor
- ◆ `smo.log`
- ◆ Shared memory maximum
- ◆ Swap space information
- ◆ Memory information
- ◆ Kernel parameters
- ◆ FSTAB

General steps for creating a dump file from the CLI

This section takes you through the basic steps of creating a dump file. For a syntax example, see [“Creating a dump file from the CLI”](#) on page 119.

The basic steps for creating a dump file are as follows:

1. (Optional) Specify the name of the operation for which you want information. This name can be either:
 - ❖ The label you supplied when you performed the operation
 - ❖ The GUID
2. (Optional) Specify how you want SnapManager to handle messages. By default, SnapManager displays the error and warning messages on the console. To change the behavior, include either the `-quiet` option or the `-verbose` option on the command line. For more information on these options, see [“Common options for handling messages shown on the console”](#) on page 136.

Creating a dump file from the CLI

For details on the options and arguments used in the commands to get dump information, see [“The smo profile dump command”](#) on page 172, the [“The smo operation dump command”](#) on page 163, or the [“The smo system dump command”](#) on page 188.

Creating Diagnostic information from the GUI

To create the same type of information file about a failed backup, clone, or operation from the GUI, follow these steps.

Follow this procedure to create diagnostic information from the GUI.

1. Select the failed operation.
2. Right click and select **Create Diagnostics**. The Welcome to SnapManager for Oracle Diagnostic Creation Wizard opens. Click **Next**.
3. At the Diagnostic Configuration Information screen, the host for your database repository information appears.
4. To create diagnostic information without specifying an operation, select **No operation**.
5. To create diagnostic information for a specific operation, select **Operation Label**.
6. To create diagnostic information for a specific operation ID, select **Operation ID**.
7. Click **Next**.
8. At the Commit Changes screen, click **Create** to create the diagnostic information.
9. Click **Finish** if you successfully created the diagnostics. Click the **Operation Details** to see the status of the operation and the location of the .jar file. Click **Back** to correct any information.

General troubleshooting tips for using the GUI

The backups are not visible in the GUI

Symptom: When you restart the GUI and try to check the backups for a certain profile, it looks as if all the backups have disappeared. All you see are the names of the profiles.

Explanation: Each time you start a GUI session, you must authenticate the profile. SnapManager does not display any information about the profile until you authenticate it.

What to do: Right-click on the profile and select the **Authenticate** option from the pop-up menu. SnapManager displays the Profile Authentication dialog box. Enter the database user name and password. Once you do this, SnapManager displays the backup list.

You only need to authenticate the profile once during each GUI session.

General troubleshooting tips for working with clones

Avoid using a duplicate SID for the clone name

If you have trouble creating a clone, make sure you are supplying a unique SID for the clone.

SnapManager only supports cloning a database on the same host. Because Oracle does not permit you to run two databases with the same SID simultaneously on the same host, you must specify a new SID for the clone.

Clone attempt fails saying mountpoint is in use

Symptom: The clone operation fails with a message saying that the mount path you are using is already in use.

Explanation: SnapManager does not let you mount a clone over an existing mountpoint. This means it is possible that an incomplete clone did not remove the mountpoint.

What to do: Run the `smo operation abort` command with the clone SID as the label. Doing this releases the resources so you can use them again.

For example, suppose you tried unsuccessfully to create a clone using the following command:

```
$ smo clone create -newsid racspcl -profile SMORMANDBASE -label CLONETP
```

You would then run the following `smo backup delete` command:

```
$ smo operation abort clone -label racspcl -profile SMORMANDBASE
```

You can use the `ls` command with the mountpoint. If it no longer exists, you should get a message saying “no such file or directory.”

NID-00100 error occurs during clone operation

Symptom: The clone operation fails with an NID-00100 error message.

Explanation: Some versions of Oracle’s NID utility do not work with data files unless the files use a `.dbf` extension.

What to do: If you encounter a an NID-00100 error during a clone operation, complete the following steps.

Step	Action
1	Rename the data file to give it a .dbf extension.
2	Repeat the backup operation.
3	Now clone the new backup.

General troubleshooting tips for SnapManager

Backups registered in RMAN are not visible in the backup list

Symptom: The `backup list` command that is part of RMAN does not display any of the backups that are registered with RMAN.

Explanation: SnapManager backups that are registered with RMAN don't display when you use the `backup list` command.

What to do: Execute the `list` command instead.

RMAN retention policy and SnapManager for Oracle conflicts

Symptom: SnapManager for Oracle gives an error message indicating it cannot delete a backup from RMAN.

Explanation: RMAN may have its own retention policy for data file backups that SnapManager for Oracle registers with RMAN. RMAN may delete the backup information from its catalog while SnapManager for Oracle still has the backup in a repository.

What to do: Use the `-force` option to delete the backup.

Recovering control files takes a long time

Symptom: You attempted to perform a control file recovery; however, it is taking an exceptionally long time.

Explanation: When SnapManager uses control files to perform a recovery, it follows the Oracle requirement of having the last redo log from each redo log thread be applied in the recovery process. SnapManager attempts all permutations of redo logs from the different threads until one succeeds. As the number of threads increases, the amount of time the recovery takes also increases.

What to do: If you know which redo logs are needed, it might be quicker to manually recover the database.

Deleting a profile containing an incomplete operation leaves a SnapManager lock file

Symptom:: You received an error message when attempting to delete a profile that there are incomplete operations.

Explanation: If a profile contains an incomplete operation and you attempt to remove the profile, SnapManager does not remove the lock on the target Oracle database. SnapManager will display an error similar to:

SMO-05035 [ERROR]: SMO-05035: Cannot delete profile "GSF1BDEL" because it contains 1 incomplete operation(s): 8abc01130e529a75010e529aa80a0001. To delete the profile, you must either wait for the incomplete operation(s) to finish or you must abort the operation(s) with the "smo operation abort" command.

SMO-13032 [ERROR]: SMO-13032: Cannot perform operation: Profile Delete. Root cause: SMO-05035: Cannot delete profile "GSF1BDEL" because it contains 1 incomplete operation(s): 8abc01130e529a75010e529aa80a0001. To delete the profile, you must either wait for the incomplete operation(s) to finish or you must abort the operation(s) with the "smo operation abort" command.

What to do:: Issue the `smo operation abort` command to clean up failed operations or wait for the incomplete operations to finish. Remove all INCOMPLETE operations for a profile before removing a profile.

Operations involving raw LUNS on qtrees fail

Symptom: Any attempt to create an ASM backup fails when the database is using a raw LUN on a qtree as storage. In addition, clone operations fail if there is a raw LUN on a qtree.

Note

A raw LUN is one that is used either as an ASM disk or a raw device. This problem does not occur when the database is stored on a SnapDrive for UNIX-managed file system on a LUN residing on a qtree.

Explanation: SnapManager does not support raw LUNs that are on qtrees.

What to do: Reconfigure your database storage so that LUNs are created at the root volume level instead of on qtrees.

Backup operation hangs indefinitely

Symptom: The backup operation hangs when the storage system is down and cluster failover is disabled.

Explanation: SnapManager does not use a time-out mechanism for its external commands. As a result, it could hang indefinitely if Oracle or SnapDrive for UNIX hangs.

What to do: If SnapManager appears to be hanging, use the `smo operation abort` command to end the operation gracefully.

Backup operation fails without an error message on the console

Symptom: The backup operation fails, but SnapManager does not display an error message to the console.

Explanation: Check the log file for an error. Also, check the length of the comment for the backup. SnapManager enables a maximum of 255 characters in a comment. If you exceed that number of characters, the operation fails.

What to do: Limit your comments to 255 characters or less. If you attempted a backup with a comment that contained more than 255 characters, use the `smo backup delete` command to remove the failed backup. Then repeat the backup with a shorter comment.

File system containing log file is full

Symptom: SnapManager reports that the file system containing the log file is full.

Explanation: SnapManager does not use a time-out mechanism for its external commands. As a result, it could hang indefinitely if Oracle or SnapDrive for UNIX hangs.

SnapManager writes messages about the operations that it is performing to a log file. If the file system containing the log file becomes full, SnapManager may report an error similar to:

```
log4j:ERROR Failed to flush writer, java.io.IOException: No space left on device
```

This error does not affect the execution of the SnapManager operation. The operation continues, and SnapManager displays messages to the console as usual. It exits with the correct exit status

What to do: Find out which file system is full by running the `df` command. Free up space on that file system.

Cannot connect to database - invalid username password

Symptom: SnapManager for Oracle username and password failed.

Explanation: SnapManager for Oracle requires you have `sysdba` privileges when attempting commands. If the user authentication information used does not have these privileges, you will receive the error message:

```
SMO-12105 [ERROR]: Cannot connect to database - invalid
username/password.
```

Oracle error is: ORA-01017: invalid username/password; logon denied

What to do: First, verify the user has `sysdba` privileges. Perform the following steps:

1. Log in to SQLPlus as the Oracle user "sys".
2. Run the following sql command:

```
SQL> select * from v$pwfile_users;
```

USERNAME	SYSDBA	SYSOPER
SYS	TRUE	TRUE
SCOTT	TRUE	FALSE

If the user is not listed, you must enable authentication of an administrative user. To enable authentication of an administrative user using password file authentication you must do the following:

1. If not already created, create the password file using the `ORAPWD` utility:

```
ORAPWD FILE=filename PASSWORD=password ENTRIES=max_users
```
2. Set the `REMOTE_LOGIN_PASSWORDFILE` initialization parameter to `EXCLUSIVE`. (This is the default).

Note

`REMOTE_LOGIN_PASSWORDFILE` is a static initialization parameter and therefore cannot be changed without restarting the database.

3. Connect to the database as user `SYS` (or as another user with the administrative privileges).
4. If the user does not already exist in the database, create the user.
5. Grant the `SYSDBA` or `SYSOPER` system privilege to the user:

```
GRANT SYSDBA to oe;
```

This statement adds the user to the password file, thereby enabling connection as `SYSDBA`.

- 6.** Attempt to connect to the database as the user just created.

Common error conditions

Error conditions

This section provides general information about the common types of error messages that you may encounter. This chapter does not include information for every component or error condition.

The following table provides information about the numerical ranges for the different types of messages you might see with SnapManager.

Group	Range	Usage
ENVIRONMENT	1000-1999	Message used to log the state or issues with the operating environment of SnapManager. These include messages about systems that the SnapManager interacts with, such as the host, the storage system, database, SnapDrive for UNIX, and so on.
BACKUP	2000-2999	Messages associated with the database backup process.
RESTORE	3000-3999	Messages associated with the database restore process.
CLONE	4000-4999	Messages associated with the database clone process.
PROFILE	5000-5999	Messages associated with managing profiles.
MANAGE	6000-6999	Messages associated with managing backups.
VIRTUAL DATABASE INTERFACE	7000-7999	Messages associated with the Virtual Database Interface. These messages deal with the interface from the product to the various Oracle versions.
VIRTUAL STORAGE INTERFACE	8000-8999	Messages associated with the Virtual Storage Interface. This deals with the interface from the product to the various Data ONTAP and SnapDrive for UNIX versions.
REPOSITORY	9000-9999	Messages associated with the Repository interface.
METRICS	10000-10999	Messages associated with the size of the database backup, the elapsed time to perform the backup, the time to restore the database, the number of times a database has been cloned, and so on.

Group	Range	Usage
VIRTUAL HOST INTERFACE	11000-11999	Messages associated with the Virtual Host Interface. This is the interface to the host operating system. It hides the differences between host platforms.
EXECUTION	12000-12999	Messages associated with the execution package, including spawning and processing operating system calls.
PROCESS	13000-13999	Messages associated with the process component of SnapManager.
UTILITIES	14000-14999	Messages associated with SnapManager utilities, global context, and so on.
DUMP	15000-15999	Messages associated with dump operations.
HELP	16000-16999	Messages associated with help
SERVER	17000-17999	Messages associated with smo server administration

Where to go for more information

Relevant documentation

This guide provides information about the basic tasks involved in installing and using SnapManager. For more information, check the following on the NetApp on the Web (NOW) site:

- ◆ *SnapManager Description Page*

This page contains information about SnapManager, pointers to online documentation, and a link to the SnapManager Download page, where you can download the software. For information on getting to this page, see “[Where to find the software](#)” on page 44.

- ◆ *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products*

This document is available at http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/. It is a dynamic, online document that contains the most up-to-date information about the requirements for setting up a system in a NetApp SAN environment. It provides the current details about storage systems and host platforms, cabling issues, switch issues, and configurations.

- ◆ *SnapManager & SnapDrive Compatibility Matrix*

This document is available at <http://now.netapp.com/NOW/knowledge/docs/docs.shtml> in the Interoperability section. It is a dynamic, online document that contains the most up-to-date information specific to SnapManager and its platform requirements. (When you go to the SnapManager & SnapDrive Compatibility Matrix main page, select the link to the SnapManager for Oracle matrix.)

- ◆ *SnapManager Release Notes*

This document comes with SnapManager. You can also download a copy from <http://now.netapp.com>. It contains any last-minute information that you need to get the configuration up and running smoothly.

SnapManager requires other NetApp products. As a result, the following documentation about both SnapManager and the NetApp products it works with might also be useful to you:

- ◆ The documentation that comes with SnapDrive for UNIX, including the SnapDrive for UNIX Compatibility Matrix. This matrix is available at <http://now.netapp.com/NOW/knowledge/docs/docs.shtml> in the Interoperability section.

Download these documents from the NOW site. See the SAN/IPSAN library at <http://now.netapp.com/NOW/knowledge/docs/san/>. Click Snapdrive to find:

- ❖ *SnapDrive for UNIX Release Notes*
- ❖ *SnapDrive for UNIX Installation and Administration Guide (IBM AIX, HP-UX, Linux, Solaris)*
- ❖ *SnapDrive for UNIX Quick Start Guide (IBM AIX, HP-UX, Linux, Solaris)*
- ◆ The documentation that comes with the NetApp host attach and support kits.
Download these documents from the NOW site. See the SAN/IPSAN library at <http://now.netapp.com/NOW/knowledge/docs/san/>.
- ◆ *System Configuration Guide*
This document is available at <http://now.netapp.com/NOW/knowledge/docs/docs.shtml/>. This is an online document that contains information about the supported storage system models for Data ONTAP.
- ◆ *Data ONTAP Block Access Management Guide*
This document is available from the Data ONTAP library on the NOW site at <http://now.netapp.com/NOW/knowledge/docs.shtml/>. It provides information about using Data ONTAP and setting up your storage system to work with Data ONTAP.
- ◆ Technical reports on using NetApp products with Oracle.
There are technical reports available on the www.netapp.com site as well as with vendors who provide NetApp and Oracle products.
- ◆ Host operating system and Oracle database information
NetApp does not provide these documents. See the readme files and other documentation that you received with your host operating system.

About this chapter

This chapter provides details about the CLI commands to use with SnapManager.

This chapter covers the following topics:

- ◆ “[Collecting information needed by SnapManager Commands](#)” on page 134
- ◆ “[Common options for handling messages shown on the console](#)” on page 136
- ◆ “[The smo backup create command](#)” on page 137
- ◆ “[The smo backup delete command](#)” on page 142
- ◆ “[The smo backup list command](#)” on page 144
- ◆ “[The smo backup mount command](#)” on page 146
- ◆ “[The smo backup restore and recovery command](#)” on page 148
- ◆ “[The smo backup unmount command](#)” on page 151
- ◆ “[The smo backup verify command](#)” on page 153
- ◆ “[The smo clone create command](#)” on page 155
- ◆ “[The smo clone template command](#)” on page 157
- ◆ “[The smo cmdfile command](#)” on page 158
- ◆ “[The smo operation abort command](#)” on page 161
- ◆ “[The smo operation dump command](#)” on page 163
- ◆ “[The smo profile create command](#)” on page 165
- ◆ “[The smo profile delete command](#)” on page 170
- ◆ “[The smo profile dump command](#)” on page 172
- ◆ “[The smo profile list command](#)” on page 173
- ◆ “[The smo profile show command](#)” on page 174
- ◆ “[The smo profile update command](#)” on page 176
- ◆ “[The smo profile verify command](#)” on page 181
- ◆ “[The smo repository create command](#)” on page 182
- ◆ “[The smo repository delete command](#)” on page 184
- ◆ “[The smo repository update command](#)” on page 186
- ◆ “[The smo system dump command](#)” on page 188
- ◆ “[The smo system verify command](#)” on page 189
- ◆ “[The smo version command](#)” on page 190

Collecting information needed by SnapManager Commands

Command information

The following information is included for each command in this section:

- ◆ Valid usage formats
- ◆ Information on the keywords, options, and arguments available with the commands and the values you should supply
- ◆ Examples of the commands

General notes about the commands

The following are general notes about the commands:

- ◆ It is recommended that you supply a name for the backup using the `-label` option when you create a backup. This provides you with an easy way to identify the backup.

Labels are case-sensitive. When you use the label, make sure you enter it exactly as you created it.

If you do not specify a label, SnapManager creates a default label.

- ◆ SnapManager automatically generates a Globally Unique Identifier (GUID) for each backup, which is a 32-character HEX string.

To determine what the GUID is, run the `smo backup list` command.

- ◆ If the shell on your host limits the number of characters that can appear on a command line, you can use the `cmdfile` command. This command lets you put a SnapManager command in a text file and execute it that way. For more information, see “[The smo cmdfile command](#)” on page 158.

- ◆ Get information about individual commands by issuing the `smo help` command. This command has the following format:

```
# smo help  
[backup|clone|cmdfile|operation|profile|repository|system|version]
```

- ◆ Specify how you want messages displayed. For details on the options to control message output, see “[Common options for handling messages shown on the console](#)” on page 136.

Conventions for entering names and command options

Command options are case-sensitive

SnapManager command options are case-sensitive. For example, if you enter a label name, use the same case structure that you used when you created. For example, if you create the label “WedTest”, make sure you enter the label exactly that way each time you use it.

Avoid spaces in names

SnapManager does not accept spaces in:

- ◆ Profile or label names
- ◆ File names or file paths

Note

If your target database contains spaces in the file names or paths to the files, reconfigure your file system layout to remove the spaces.

In addition, do not use spaces to create a location for the clone in the clone specification file.

Limits for the number of characters used

SnapManager accepts comments up to 200 characters and labels up to 80 characters.

Common options for handling messages shown on the console

Command options affecting the display of messages

SnapManager lets you specify how to display messages.

SnapManager has three levels of messages that it can display to the console:

- ◆ Error
- ◆ Warning
- ◆ Informational

Note

Regardless of the level of detail specified with a command, SnapManager always writes all messages types to the log files.

There are two command line options that control the amount of output SnapManager displays on the console:

- ◆ `-quiet`. This option tells SnapManager to only display error messages to the console.
- ◆ `-verbose`. This option tells SnapManager to display error, warning, and informational messages to the console.

If you do not specify either option, SnapManager displays error messages and warnings to the console.

The smo backup create command

Using the smo backup create command

The `smo backup create` command lets you create backups of databases on one or more storage system volumes. When the `smo backup create` command completes, it returns a Globally Unique Identifier (GUID) for the backup. The GUID is a 32-character HEX string generated by the system. (For more information about the GUID, see [“General notes about the commands”](#) on page 134.)

Usage format for smo backup create:

```
smo backup create
    {-auto | -online | -offline}
    {-full | -files list | -tablespaces list}
    -profile profile_name
    [-verify]
    [-label name]
    [-comment comment]
    [-force]
    [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo backup create` command.

Note

Before you perform this command, you must create a database profile using the `smo profile create` command. For information on how to do this, see [“The smo profile create command”](#) on page 165.

Requirement	Argument
Specify the state the database must be in when you perform the backup. ◆ <code>-auto</code>. SnapManager checks the state of the database and determines which kind of backup to perform: ❖ If the archive log mode and database online mode are enabled, it performs an online backup of the database. ❖ If the database is in a mounted state, it performs an offline backup. ❖ If the <code>-force</code> option is specified and none of the above are true, it does an offline backup ◆ <code>-online</code>. SnapManager performs the backup while the database is online. ◆ <code>-offline</code>. SnapManager performs the backup while the database is shut down. In this case, the database is mounted, but not open. If the database is up when you include this option, the backup fails. Note_____ If you specify the <code>-online</code> option when the database is shut down or the <code>-offline</code> option when the database is up, the backup fails. Use the <code>-force</code> option to have SnapManager attempt to alter the database state before making the backup. _____	
◆ <code>-online</code>	~
◆ <code>-offline</code>	~
◆ <code>-auto</code>	~
Specify whether you want SnapManager to back up the entire database or only part of it: ◆ When you specify a full backup, SnapManager makes a backup of the entire database. This includes all the data, log and control files. ◆ When you specify a partial backup, SnapManager backs up only the portion of the database you specify and the log and control files. Specify either a list of tablespace names (<code>-tablespaces</code>) or a list of file names (<code>-files</code>). Use spaces to separate the names in the list. Do not include any spaces in these lists. If you specify file names and the database is up, SnapManager ensures that the appropriate tablespaces are for online backup mode.	
◆ <code>-full</code>	~
◆ <code>-files list</code>	<i>List of file names separated by spaces</i>
◆ <code>-tablespaces list</code>	<i>List of tablespace names separated by spaces</i>

Requirement	Argument
Provide information about the database by entering the name of the profile for the database you want to back up. Note_____ You must have set up this information previously using the <code>smo profile create</code> command. See the “The smo profile create command” on page 165. _____	
◆ <code>-profile name</code>	<i>name of the profile</i>
(Optional) Specify whether you want to SnapManager to verify that the files in the database were not corrupt. If you include the <code>-verify</code> option, SnapManager invokes the Oracle utility <code>dbv</code>. Note_____ If you include this option, the backup operation does not complete until the verify operation completes. _____ For more information about separately verifying a backup, see the “Using the smo backup verify command” on page 153.	
◆ <code>-verify</code>	~
◆ <code>-label name</code>	<i>unique name you want to assign to the backup</i>

Requirement	Argument
(Optional) Specify information about the backup. Enter the following information: ◆ <code>-label</code>. Enter a name for the backup. This name must be unique for that profile. This option lets you supply an easy-to-recognize name for the backup. The name you supply can contain alphanumeric characters and the following special characters: underscore (<code>_</code>) and hyphen (<code>-</code>). It cannot start with a hyphen. If you do not supply a label, SnapManager creates a default label that has the format <code><scope>_<type>_<date></code> where: ❖ <code>scope</code> is either F to indicate a full backup or P to indicate a partial backup. ❖ <code>type</code> is either C to indicate an offline (cold) backup or H to indicate an online (hot) backup. ❖ <code>date</code> is the year, month, day, and time of the backup. SnapManager uses a 24-hour clock. For example, if you performed a full backup with the database offline on Jan. 16, 2006, at 5:45:16 p.m. Eastern standard time, SnapManager would create the label <code>F_C_20060116174516EST</code>. Whether you specify a label or not, SnapManager generates a GUID for the backup. (See “General notes about the commands” on page 134.) ◆ <code>-comment</code>. Supply a comment to describe this backup. This comment can be any single-quoted string. For example, you might enter ‘QA database’ so that you can quickly identify the contents of the backup. Note _____ Some shells strip off quote marks. If that is true for your shell, you must escape the quote with a backslash (<code>\</code>). For example, you might need to enter: <code>\ ' this is a comment \ '</code> to keep your shell from stripping off the quote marks. _____	
(Optional) Decide if you want SnapManager to force a state change if the database is not in the correct state. For example, if you use this option, SnapManager might change the state of the database from online to offline, based on the type of backup you specify and the state that the database is in.	
◆ <code>-force</code>	~
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136. For details on these options, see “Common options for handling messages shown on the console” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Example of smo backup create command line

The following is a sample command line that uses the `smo backup create` command.

```
# smo backup create -full -offline -profile SMOKE_SMOASM_BISMARCK -  
label smoke_full_backup -force
```

Output:

Operation Id [8abc01ec0e79356d010e793581f70001] succeeded.

The smo backup delete command

Using the smo backup delete command

The `smo backup delete` command lets you remove backups. Use this to prune your backup list or to delete backups that SnapManager does not automatically remove, such as backups that were used to create a clone or backups that failed.

Use this command to delete backups that are not affected by the `-retain` option set when you created or modified the profile (see “[The smo profile create command](#)” on page 165 and “[The smo profile update command](#)” on page 176.)

Usage format for smo backup delete:

```
smo backup delete
  {-id guid | -label label_name | -retain number | -all }
  -profile profile_name
  [-force]
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo backup delete` command.

Requirement	Argument
Decide whether you want to delete a specific backup or limit the number of backups you have to a specific number.	
To delete a specific backup enter one of the following:	
◆ <code>-id <i>guid</i></code>. The GUID that SMO generated when you created the backup. (For more information about GUIDs, see “General notes about the commands” on page 134.) ◆ <code>-label <i>label_name</i></code>. The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup create</code> command line.	
To delete successful backups until you have only a specific number remaining, enter the following:	
◆ <code>-retain <i>number</i></code>. This option tells SnapManager how many backups to keep. It keeps the number of backups you specify and removes all other backups. It removes the oldest backups first. For example, if you had five backups labeled May, June, July, August, and September and you put <code>-retain 3</code> on the command line, SnapManager would remove the backups May and June. If you specify 3 backups to be retained and you only have 2 backups, SnapManager does not remove any backups. To delete all the backups, set this value to 0. ◆ <code>-all</code>. This tells SnapManager to delete all backups.	
◆ <code>-id <i>guid</i></code>	<i>GUID generated when you created the backup</i>

Requirement	Argument
◆ <code>-label <i>label_name</i></code>	<i>name you supplied when you created the backup</i>
◆ <code>-retain <i>number</i></code>	<i>number of backups to keep; if you enter 0, SnapManager removes all backups</i>
Specify the name of the profile associated with the backup you want to remove.	
◆ <code>-profile <i>profile_name</i></code>	<i>The name of the profile from which the backup was made</i>
(Optional) Decide if you want SnapManager to force the removal of the backup. When you include this option, SnapManager removes the backup even if there are problems freeing the resources associated with the backup. For example, if the backup was cataloged with RMAN, but the RMAN database no longer exists, including <code>-force</code> on the command line lets SnapManager delete the backup even though it cannot connect with RMAN.	
◆ <code>-force</code>	~
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “ Common options for handling messages shown on the console ” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo backup delete command line

The following are sample command lines that use the `smo backup delete` command.

```
# smo backup delete -profile SMOKE_NFSTEST_BISMARCK -label
smoke_online_files
```

Output:

```
Operation Id [8abc01ec0e79004b010e79006da60001] succeeded.
```

The smo backup list command

Using the smo backup list command

The `smo backup list` command displays information about backups.

Usage format for smo backup list:

```
smo backup list -profile profile_name
```

The following is a summary of the information you need to supply when you use the `smo backup list` command.

Requirement	Argument	Value you want to supply
Specify the name of the profile that contains this backup.		
◆ <code>-profile <i>profile_name</i></code>	<i>name of the profile</i>	

Example of smo backup list command line

The following is a sample command line that uses the `smo backup list` command.

```
# smo backup list -profile SMOKE_NOLVMDDB_BISMARCK
```

Output:

```
Backup id: 8abc01ec0e79083e010e79084c970001
Status: SUCCESS
Label: smoke_full_backup
Comment:
Start time: 2006-10-24 02:35:49
End time: 2006-10-24 02:40:01
Verification Status: OK
Database: NOLVMDDB
  Tablespace: USERS
    Datafile: /mnt/weever/bismarck_san_smoke/NOLVMDDB/users01.dbf
[ONLINE]
  Tablespace: SYSAUX
    Datafile: /mnt/weever/bismarck_san_smoke/NOLVMDDB/sysaux01.dbf
[ONLINE]
  Tablespace: SYSTEM
    Datafile: /mnt/weever/bismarck_san_smoke/NOLVMDDB/system01.dbf
[SYSTEM]
  Tablespace: UNDOTBS1
```

```
Datafile: /mnt/weever/bismarck_san_smoke/NOLVMDb/undotbs01.dbf
[ONLINE]
Host: bismarck
Filesystem: /mnt/weever/bismarck_san_smoke/NOLVMDb
Storage System: weever.rtp.netapp.com
Volume: bismarck_san_smoke
Snapshot:
SMO_SMOKE_NOLVMDb_BISMAR_NOLVMDb_F_C_0_8abc01ec0e79083e010e79084c9
70001_0
```

The smo backup mount command

Using the smo backup mount command

Note

The smo backup mount command lets you mount a backup in order to perform a recover operation using RMAN. You should only need to use this command if you are using an external tool such as RMAN. SnapManager automatically handles mounting backups if you use the smo backup restore command to restore the backup. This command displays a list of paths where the snapshots in the backup have been mounted.

Usage format for smo backup mount:

```
smo backup mount
    {-id guid | -label label_name}
    -profile profile_name
    [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the smo backup mount command.

Requirement	Argument
Specify the name of the backup that you want mounted. This can be either of the following:	
◆ The default GUID that SMO generated when you created the backup. This is a 32-character HEX string.	
◆ The name you specified when the backup was created. You have the option of entering a name as the value for -label when you set up the smo backup create command line.	
◆ -id guid	GUID generated when you created the backup
◆ -label label_name	name you supplied when you created the backup
Specify the name of the profile associated with the backup.	
◆ -profile profile_name	name of the profile
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ -quiet	~
◆ -verbose	~

Example of smo backup mount command line

The following is a sample command line that uses the `smo backup mount` command.

```
# smo backup mount -label EXC79-Prod-Full -profile EXC79
```

The smo backup restore and recovery command

Using the smo backup restore command

The smo restore command lets you restore backups of a database or a portion of a database and then recover the database information.

Usage format for smo restore:

```
smo backup restore
  -profile profile_name {-label label_name | -id guid}
  {-complete | -files list | -tablespaces list | -controlfiles}
  [-recover -nologs | -alllogs | -until {scn | date}]
  [-using_backup_controlfile]
  [-force] [-verbose]
```

The following is a summary of the information you need to supply when you use the smo backup restore command.

Requirement	Argument
Specify the name of the profile associated with this backup.	
◆ -profile <i>profile_name</i>	<i>name of the profile</i>
Specify the name of the backup that you want to restore. This can be either of the following: ◆ The default GUID that SMO generated when you created the backup. ◆ The name you specified when the backup was created. You have the option of entering a name as the value for -label when you set up the smo backup create command line.	
◆ -label <i>label_name</i>	<i>name you supplied when you created the backup</i>
◆ -id <i>guid</i>	<i>GUID generated when you created the backup</i>
Specify whether you want SnapManager to restore the entire database or only part of it: ◆ When you specify the complete option, SnapManager restores everything in the backup. ◆ When you specify a list of tablespace names (-tablespaces) or a list of file names (-files), SnapManager restores only those tablespaces or files. Use spaces to separate the names in the list. If you specify filenames and the database is up, SnapManager ensures that the tablespace containing the files is in offline. ◆ When you specify the control files option, SnapManager restores only the control files. You cannot specify -controlfiles with -complete.	
◆ -complete	~

Requirement	Argument
◆ -files <i>list</i>	<i>a list of file names separated by spaces</i>
◆ -tablespaces <i>list</i>	<i>list of tablespace names separated by spaces</i>
◆ -controlfiles	~
(Optional) If you want SnapManager to recover, include the <code>-recover</code> option. Then specify the point in time to which you want SnapManager to recover the database. SnapManager provides the following options based on whether you want: ◆ <code>-nologs</code> SnapManager recovers the database to the time of the backup. ◆ <code>-alllogs</code> SnapManager recovers the database using all available logs. ◆ Specify <code>-until</code> along with either date or scn. ❖ <i>date</i>. SnapManager recovers the database up to the date and time you specify. Use the format year:month:date:hour:minute:second (yyyy-mm-dd:HH:MM:SS). For example, you might enter 2006-01-25:19:06:22 to specify that the backup should be restored up to Nov. 11, 2005 at 7:06 p.m. 22 seconds. (Whether you use a 24-hour clock or not depends on how you’ve set up the options for your Oracle database.) Note _____ This is the time associated with the database. It is not necessarily the clock time on the host. _____ ❖ <i>scn</i>. SnapManager rolls forward the Oracle SCN log files until it reaches the SCN number you specify. For example, you might enter 384641. SnapManager would recover the backup up to that SCN.	
(Optional) Decide if you want SnapManager to change the database state to a lower state than its current state. For RAC, you must include the <code>-force</code> option if SnapManager needs to change the state of any RAC instance to a lower state. By default, SnapManager can change the database state to a higher state during an operation. You do not need to enter this option for SnapManager to change the database to a higher state.	
◆ -force	~
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ -quiet	~
◆ -verbose	~

Example of smo backup restore command line

The following are sample command lines that use the `smo backup restore` command.

```
# smo backup restore -files  
/mnt/triggerfish/bt_bismarck_nfs_smoke/datafiles/users01.dbf  
-label smoke_online_files -profile SMOKE_NFSTEST_BISMARCK -recover  
-alllogs
```

Output:

Operation Id [8abc01ec0e78f3e2010e78f3fdd00001] succeeded.

The smo backup unmount command

Using the smo backup unmount command

Note

The `smo backup unmount` command lets you unmount a backup. You should only need to use this command if you are using an external tool such as RMAN.SnapManager automatically handles unmounting backups if you use the `smo backup restore` command. You can use this command to free up resources on the host when you no longer need the backup mounted.

Usage format for smo unmount:

```
smo backup unmount
  {-id guid | -label label_name}
  -profile profile_name
  [-force]
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo backup unmount` command.

Requirement	Argument
Specify the name of the backup you want to unmount. This can be either of the following:	
◆ The default GUID that SMO generated when you created the backup. This is a 32-character HEX string. ◆ The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup create</code> command line.	
◆ <code>-id <i>guid</i></code>	<i>GUID generated when you created the backup</i>
◆ <code>-label <i>label_name</i></code>	<i>name you supplied when you created the backup</i>
Specify the name of the profile associated with this backup.	
◆ <code>-profile <i>profile_name</i></code>	<i>name of the profile</i>
(Optional) Decide if you want SnapManager to force an unmount.	
Note When you include this option, SnapManager performs an unmount even if there are problems freeing the resources associated with the backup. SnapManager for Oracle tries to unmount the backup and clean up any associated resources. The log shows the unmount operation as successful, but you may have to manually clean up resources if there were errors in the log.	

Requirement	Argument
◆ -force	~
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “ Common options for handling messages shown on the console ” on page 136.	
◆ -quiet	~
◆ -verbose	~

Examples of smo backup unmount command line

The following is a sample command line that uses the `smo backup unmount` command.

```
# smo backup unmount -label EXC79-Prod-Full -profile EXC79
```

The smo backup verify command

Using the smo backup verify command

The `smo backup verify` command lets you check to see if the backup is in a valid format for Oracle. It checks the block structure within the backed up database. You can execute this command at any time; you do not need to execute it at the same time you perform the backup operation.

Note—————
If you use the `-verify` option with the `smo backup create` command, it automatically executes the command after it completes the backup. In fact, the backup operation does not complete until the verify option finishes.

The `smo verify` command calls the Oracle `dbv` command for each data file in the backup.

Usage format for smo verify:

```
smo backup verify {-id guid | -label label_name}
                  -profile profile_name [-force]
```

The following is a summary of the information you need to supply when you use the `smo backup verify` command.

Requirement	Argument
Specify the name of the backup that you want to verify. This can be either of the following:	
◆ The default GUID that SMO generated when you created the backup. This is a 32-character HEX string.	
◆ The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup create</code> command line.	
◆ <code>-id guid</code>	<i>GUID generated when you created the backup</i>
◆ <code>-label label_name</code>	<i>name you supplied when you created the backup</i>
Specify the name of the profile that contains this backup.	
◆ <code>-profile profile_name</code>	<i>name of the profile</i>
(Optional) Decide if you want SnapManager to force the verify operation. If you include this option, it forces the database into the necessary state to perform the verify operation.	
◆ <code>-force</code>	<i>~</i>

Example of smo backup verify command line

The following is a sample command line that uses the `smo backup verify` command.

```
# smo backup verify -profile SMOKE_SMOASM_BISMARCK -label  
smoke_full_backup -quiet
```

Output:

```
DBVERIFY - Verification starting : FILE =  
+SMO_1_1161675083835/smoasm/datafile/smoke.277.582482539  
...
```

The smo clone create command

Using the smo clone create command

The `smo clone create` command lets you create a clone of a backed up database. You can only use this command with full backups.

Usage format for smo clone create:

```
smo clone create
  {-id guid | -label label_name}
  -current
  -newsid new_sid
  -profile profile_name
  -clonespec file_name
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo clone create` command.

Requirement	Argument
Specify the backup containing the database you want to clone. The backup must be a full, offline backup; you cannot clone a partial backup. To specify the backup, supply one of the following:	
◆ The default GUID that SMO generated when you created the backup. This is a 32-character HEX string. ◆ The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup create</code> command line.	
◆ <code>-id <i>guid</i></code>	<i>GUID generated when you created the backup</i>
◆ <code>-label <i>label_name</i></code>	<i>name you supplied when you created the backup</i>
◆ <code>-current</code>	<i>backup and clone from the current state of the database</i>
Specify a new Oracle SID for the clone to use. This identifier can be a maximum of eight characters.	
Note _____	
Oracle does not let you run two databases with the same SID simultaneously on the same host.	
◆ <code>-newsid <i>new_sid</i></code>	<i>Oracle SID for the cloned database</i>
Specify the name of the profile associated with this backup.	
◆ <code>-profile <i>profile_name</i></code>	<i>name of the profile</i>

Requirement	Argument
Specify the pathname to the clone specification file for the clone. This can be a relative or absolute pathname. For information about creating a clone specification, refer to “ Creating a clone specification ” on page 80.	
◆ <code>-clonespec filename</code>	<i>path to the clone specification file</i>
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “ Common options for handling messages shown on the console ” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo clone create command line

The following is a sample command line that uses the `smo clone create` command.

```
# smo clone create -newsid CLONE -clonespec
/opt/NetApp/smo/clonespecs/SOASM_bismarck.xml -profile
SMOKE_SMOASM_BISMARCK -label smoke_full_backup
```

Output:

```
Operation Id [8abc01ec0e794e3f010e794e6e9b0001] succeeded.
```

Once the `smo clone create` command completes, the database is up and running.

The smo clone template command

Using the smo clone template command

The `smo clone template` command lets you create a clone specification of a backed up database. You can only use this command with full backups that were created while the database was offline.

Usage format for smo clone template:

```
smo clone template
  -profile name {-id guid | -label label_name}
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo clone template` command.

Requirement	Argument
Specify the profile associated with the database you want to clone. Supply one of the following:	
◆ The default GUID that SnapManager for Oracle generated for the profile. This is a 32-character HEX string.	
◆ The name you specified when the profile was created. You have the option of entering a name as the value for <code>-label</code> when you used the <code>smo profile create</code> command line.	
◆ <code>-id guid</code>	<i>GUID generated when you created the backup</i>
◆ <code>-label label_name</code>	<i>name of the backup</i>
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo clone template command line

The following is a sample command line that uses the `smo clone template` command.

```
# smo clone template -profile EXC79 -label mybackup
```

Once the `smo clone template` command completes, the clone specification template is complete. Edit the template as explained in [“Creating the clone specification”](#) on page 81.

The smo cmdfile command

Using the smo cmdfile command

Put commands in individual text files and use `smo cmdfile` to execute the command in the text file.

The `smo cmdfile` command replaces the `smo pfile` command, which is deprecated in this release and will not work.

Note

The `smo cmdfile` is not backwards compatible with the `smo pfile` command.

Use the `smo cmdfile` command if the shell on your host limits the number of characters that can appear on a command line. You can only put one command in each text file. Within the text file, drop the `smo` from the command information. See “[Example of the text file](#)” on page 158. When referencing the file from the command line, use the command and its options as `smo cmdfile -file file_name`.

Usage format for smo cmdfile:

```
smo cmdfile -file file_name
```

The following is a summary of the information you can supply when you use the `smo cmdfile` command.

Requirement	Argument
Create a text file to contain the command. You must create a new file for each command.	
Place the command line and all its options in the text file.	
Save the text file and enter its path name on the <code>smo cmdfile</code> command line. SnapManager processes the command you entered in the file.	
◆ <code>-file file_name</code>	<i>path to text file containing the command line you want to execute</i>

Example of the text file

This example creates a profile by putting the profile command in a text file.

A text file called `commands.txt` in `/tmp` has the following command in it:

```
profile create -profile SMOKE_SMOASM_BISMARCK -repository
-dbname SMOREPO -login -username bismarck_smoke_user -password
netapp -port 1521 -host bismarck -database -dbname SMOASM -sid
SMOASM -login -username sys -password oracle -port 1521 -host
bismarck -osaccount oracle
```

Example of smocmdfile command line

Now you can create the profile by entering the `smo cmdfile` command with that file.

```
# smo cmdfile -file /tmp/commands.txt
```

The smo help command

Using the smo help command

The `smo help` command displays information about the SnapManager commands and their arguments. If you do not supply an argument with this command, it displays a list of valid commands. If you supply a command with this argument, it displays the syntax graph for that command.

Usage format for smo profile help:

```
smo help [command_name]
```

The following is a summary of the information you need to supply when you use the `smo help` command.

Requirement	Argument
(Optional) Include the command about which you want information. See “ General notes about the commands ” on page 134. If you omit the this option, SnapManager displays a list of commands.	
◆ <i>command</i>	<i>SnapManager command</i>

Example of smo help command line

The following is a sample command line that uses the `smo help` command to display information about the `smo backup` command.

```
# smo help backup
```

The smo operation abort command

Using the smo operation abort command

The `smo operation abort` command tells SnapManager to halt the current operation and clean up after it. Anytime you need to stop an operation or an interrupt occurs to halt an operation, you need to execute this command.

Attention — Executing this command could put the database in an unknown state.

You can also use this command if SnapManager performed an operation on the database or storage that needs to be undone. For example, if it took a snapshot that needs to be undone, you would use the abort operation to undo the action.

SnapManager keeps a persistent state of which processes are running to prevent incompatible operations from executing simultaneously. The `smo operation abort` command enables SnapManager to perform a graceful shutdown and maintain this persistent state.

Usage format for smo operation abort:

```
smo operation abort
    {-id guid | -label label_name}
    -profile profile_name
    [-force] [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo operation abort` command.

Requirement	Argument
Specify the backup the operation you want to abort is using. You can identify the backup by either of the following:	
◆ The GUID that SMO generated when you created the backup. (For more information about GUIDs, see “General notes about the commands” on page 134.) ◆ The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup create</code> command line.	
◆ <code>-id guid</code>	<i>GUID generated when you created the backup</i>
◆ <code>-label label_name</code>	<i>name you supplied when you created the backup</i>
Specify the name of the profile associated with this backup.	

Requirement	Argument
◆ <code>-profile <i>profile_name</i></code>	<i>name of the profile</i>
7. (Optional) Decide if you want SnapManager to force the operation to abort. If use this option, you may need to perform a manual cleanup if SnapManager fails to abort all actions of the operation.	
◆ <code>-force</code>	~
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Example of smo operation abort command line

The following is a sample command line that uses the `smo operation abort` command.

```
# smo operation abort -label EXC79-Prod-Full -profile EXC7
```

The smo operation dump command

Using the smo operation dump command

The `smo operation dump` command lets you create a dump file containing information about an operation. This file can be helpful if you need to troubleshoot a problem with the operation.

Usage format for smo operation dump:

```
smo operation dump
  {-id guid | -label label_name}
  -profile profile_name
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo operation dump` command.

Requirement	Argument
◆ The default GUID that SMO generated when you created the backup. This is a 32-character HEX string. ◆ The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup create</code> command line.	
◆ <code>-id <i>guid</i></code>	<i>GUID generated when you created the backup</i>
◆ <code>-label <i>label_name</i></code>	<i>name you supplied when you created the backup</i>
Specify the name of the profile associated with this backup.	
◆ <code>-profile <i>profile_name</i></code>	<i>name of the profile</i>
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo operation dump command line

The following is a sample command line that uses the `smo operation dump` command.

```
# smo operation dump -label smoke_offline_tablespace -profile  
SMOKE_SMOASM_BISMARCK
```

Output:

```
SMO-15034: Output for the dump command is stored in file  
"/tmp/smo_dump_8abc01ec0e7926f8010e792701380001.jar".  
Dump file created. Host: bismarck Path:  
/tmp/smo_dump_8abc01ec0e7926f8010e792701380001.jar
```

The smo profile create command

Using the smo profile create command

The smo profile create command lets you create a profile of a database. This profile contains the host and database credentials SnapManager needs in order to perform its backup and restore operations.

Usage format for smo profile create:

```
smo profile create
-profile profile_name
{-repository -dbname repo_dbname -host repo_host -login
-password repo_password -username repo_username -port repo_port}
{-database -dbname dbname -host dbhost -login -password db_password
-username db_username -port db_port [-sid db_sid]}
{-osaccount account_name}
[-rman {-tnsname tnsname -login -password rman_password -username
rman_username} | -controlfile ]
[-asm -host asm_host -login -password asm_password -username
asm_username -port asm_port -sid asm_sid ]
[-retain number]
[-remove_asm]
[-remove_rman]
[-quiet | -verbose]
```

To update a profile, use the command `smo profile update` instead of `smo profile create`.

Requirement	Argument
Specify the name you want to use for this profile. This name can be up to 30 characters long and must be unique for that host.	
◆ -profile <i>profile_name</i>	<i>name of the profile</i>
Enter the name of the database. Use either the global name or the SID. You must also specify the connection information for the repository. This is the information that SnapManager needs in order to connect to the Oracle database. You must supply: ◆ The username and password for connecting to the Oracle database containing the repository. ◆ The port number to use to connect.	
◆ -repository -dbname <i>db_name</i>	<i>name of the database</i>

Requirement	Argument	
◆ -login	~	
❖ -username <i>username</i>	<i>username for connecting to the database where the repository resides</i>	
❖ -password <i>password</i>	<i>password for connecting to the database where the repository resides</i>	
◆ -host <i>host</i>	<i>name or IP address of the database host</i>	
◆ -port <i>port_number</i>	<i>port number for connecting to the database</i>	
You must specify the database to be operated on and the information SnapManager needs in order to connect to the database: ◆ The name of the database you want in this profile. ◆ The username and password for connecting to the Oracle database. ◆ The database host (either the host name or its IP address). ◆ The port number to use to connect. ◆ In some cases, you must also include the SID. By default, SnapManager uses the database name as the SID. If the SID is different from the database name, you must specify it with the <code>-sid</code> argument. For instance, if you are using RAC, you must use this argument to specify the SID of the RAC instance on the RAC node from which SnapManager is executed.		
◆ -database -dbname <i>db_name</i>	<i>name of the database being profiled</i>	
◆ -host <i>host</i>	<i>name of the database host</i>	
◆ -login	~	
❖ -username <i>username</i>	<i>username for connecting to the database</i>	
❖ -password <i>password</i>	<i>password for connecting to the database</i>	
◆ -port <i>port_number</i>	<i>port number for connecting to the database</i>	
◆ -sid <i>sid</i>	<i>SID for connecting to the database</i>	

Requirement	Argument	
(Optional) If you want to catalog the SnapManager backups with RMAN, you must include the RMAN connection information in the profile also. To use RMAN, you must supply: ◆ The option <code>-rman</code>. ◆ (Optional) The <code>-controlfile</code> option if you want to use the control files instead of the RMAN catalog. ◆ If you do not use the <code>-controlfile</code> option, you must provide the following information: ❖ The username and password for connecting to the RMAN catalog. ❖ The tnsname connection name (this is defined in the <code>tnsname.ora</code> file).		
◆ <code>-rman</code>	~	
❖ <code>-controlfile</code>	~	
◆ <code>-login</code>	~	
❖ <code>-username <i>username</i></code>	<i>username for connecting to the RMAN catalog</i>	
❖ <code>-password <i>password</i></code>	<i>password for connecting to the RMAN catalog</i>	
◆ <code>-tnsname <i>tnsname</i></code>	<i>connection name in the tnsname file</i>	
You must specify the name of the Oracle database user account. This is the account that SnapManager uses to perform the Oracle operations such as startup and shutdown. It is typically the user who owns the Oracle software on the host; for example, <code>oracle</code>.		
◆ <code>-osaccount <i>owner_name</i></code>	<i>name of the Oracle database owner account</i>	
(Optional) If you use the Oracle Automatic Storage Management (ASM) instance, you must specify the information SnapManager needs in order to connect to it: ◆ The username and password for connecting to the Oracle database. ◆ The port number to use to connect. ◆ In some cases, you must also include the SID. By default, SnapManager uses the database name as the SID. If the SID is different from the database name, you must specify it with the <code>-sid</code> argument. For instance, if you are using RAC, you must use this argument to specify the SID of the RAC instance on the RAC node from which SnapManager is executed.		
◆ <code>-asm <i>name</i></code>	<i>name of the ASM instance</i>	
◆ <code>-login</code>	~	
❖ <code>-username <i>username</i></code>	<i>username for connecting to ASM</i>	
❖ <code>-password <i>password</i></code>	<i>password for connecting to ASM</i>	

Requirement	Argument
◆ <code>-port <i>port_number</i></code>	<i>port number for connecting to ASM</i>
◆ <code>-sid <i>sid</i></code>	<i>SID for connecting to the database</i>
(Optional) This option applies only to successful backups that were not used to clone a database. It does not affect unsuccessful backups, which SnapManager keeps, or backups used to clone a database. Specify the number of successful backups you want to SnapManager to retain. Enter an integer. If you do not supply a number here, SnapManager replaces the previous backup each time it creates a new backup. Note_____ You can specify any integer you like here; SnapManager does not have a limit. If you keep every backup, you may eventually reach the snapshot limit or run out of space. _____ When you include the retain option, SnapManager keeps the most recent successful backups up to the number you specify. When SnapManager produces a successful backup that causes it to exceed this number, it deletes the oldest successful backup, unless that backup was used to clone a database. SnapManager does not delete backups used to create clones or unsuccessful backups. Use the <code>smo backup delete</code> operation to remove those backups. Note_____ The unsuccessful backups do not affect the storage. Only the successful backups require space on the storage. _____	
◆ <code>-retain <i>number</i></code>	<i>number of backups that are kept</i>
◆ <code>-remove_asm</code>	<i>remove ASM credentials from the profile</i>
◆ <code>-remove_rman</code>	<i>remove RMAN credentials from the profile</i>
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Example of smo profile create command line

The following is a sample command line that uses the `smo profile create` command.

```
# smo profile create -profile SMOKE_NFSTEST_BISMARCK -repository
-dbname SMOREPO -login -username bismarck_smoke_user
```

```
-password netapp -port 1521 -host bismarck -database -dbname  
NFSTEST -sid NFSTEST -login -username sys -password oracle -port  
1521 -host bismarck -retain 5 -osaccount oracle
```

Output:

Operation Id [8abc01ec0e78ebda010e78ebe6a40005] succeeded.

The smo profile delete command

Using the smo profile delete command

The `smo profile delete` command lets you delete a profile of a database.

Usage format for smo profile delete:

```
smo profile delete
-profile profile_name
[-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo profile delete` command.

Requirement	Argument
Specify the name you of the profile you want to delete.	
◆ -profile <i>profile_name</i>	<i>name of an existing profile</i>
(Optional) Decide if you want SnapManager to attempt to force the deletion. If the profile has backups associated with it, the delete operation fails unless you include this option. When you include the <code>-force</code> option, SnapManager attempts to delete all backups for that profile. Even with the <code>-force</code> option on the command line, the profile delete operation could fail if the database is not in the correct state. This is because SnapManager does not change the database state in order to delete backups during a profile delete. If the SnapManager cannot delete the backups in the current database state, you must either: ◆ Manually remove the backups using the <code>smo backup delete</code> command. For example, if the database is shutdown and the backups are cataloged with RMAN, you must first manually remove the backups before you can perform a successful profile delete operation. ◆ Put the database in the correct state before you perform the profile delete operation.	
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136	
◆ -quiet	~
◆ -verbose	~

Example of smo profile delete command line

The following is a sample command line that uses the `smo profile delete` command.

```
# smo profile delete -profile SMOKE_NFSTEST_BISMARCK
```

Output:

```
Operation Id [Ncaf00af0242b3e8dba5c68a57a5ae932] succeeded.
```

The smo profile dump command

Using the smo profile dump command

The `smo profile dump` command lets you create a dump file containing information about backups. If you specify a profile, SnapManager includes extra diagnostic information in the file. This file can be helpful if you need to troubleshoot a problem with the operation.

Usage format for smo profile dump:

```
smo profile dump
  -profile profile_name
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo profile dump` command.

Requirement	Argument
◆ <code>-profile <i>profile_name</i></code>	<i>name of the profile</i>
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “ Common options for handling messages shown on the console ” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo profile dump command line

The following is a sample command line that uses the `smo profile dump` command.

```
# smo profile dump -profile EXC79
```

The smo profile list command

Using the smo profile list command

The smo profile list command lets you display a list of the current profiles. If you include the `-verbose` option with this command, it displays the contents of each profile.

Usage format for smo profile list:

```
smo profile list [-verbose]
```

The following is a summary of the information you need to supply when you use the `smo profile list` command.

Requirement	Argument
(Optional) Include the <code>-verbose</code> option if you want SnapManager to display the build date and the contents of each profile. If you omit the this option, SnapManager only displays the profile descriptor.	
◆ <code>-verbose</code>	~

Example of smo profile list command line

The following is a sample command line that uses the `smo profile list` command.

```
# smo profile list
```

Output:

```
Profile name: SMOKE_NFSTEST_BISMARCK
```

The smo profile show command

Using the smo profile show command

The smo profile show command lets you display information about a profile. If you include the `-verbose` option with this command, it displays the contents of the profile you specify.

Usage format for smo profile show:

```
smo profile show -profile profile_name [-verbose]
```

The following is a summary of the information you need to supply when you use the `smo profile list` command.

Requirement	Argument
Specify the name of the profile that you want to display information on.	
◆ <code>-profile <i>profile_name</i></code>	<i>name of an existing profile</i>
(Optional) Include the <code>-verbose</code> option if you want SnapManager to display the profile contents. Without this option, SnapManager only displays the profile descriptor. If you omit the this option, SnapManager only displays the version number and the profile name.	
◆ <code>-verbose</code>	~

Example of smo profile show command line

The following is a sample command line that uses the `smo profile show` command.

```
smo profile show -profile SMOKE_NFSTEST_BISMARCK
```

Output:

```
Profile name: SMOKE_NFSTEST_BISMARCK
Target database information:
  Database name: NFSTEST
  SID: NFSTEST
  Host: bismarck
  Port: 1,521
  Username: sys
  Password:
Repository information:
  Database name: SMOREPO
```

```
SID: SMOREPO
Host: bismarck
Port: 1,521
Username: bismarck_smoke_user
Password:
RMAN information:
  Do not use RMAN
ASM information:
  Do not use ASM
Oracle user account: oracle
Backups to retain: 5
```

The smo profile update command

Using the smo profile update command

The smo profile update command lets you modify the information in an existing profile of a database. The only differences between this command and the smo profile create command are that the profile must already exist and you use update instead of create.

You must use this command if any of the user name/password information associated with the database change. If you use the command to change your user name or password, make sure you first change the information on the host or database.

When you update a profile, you only need to include the block of command options that apply to the section you are changing. For example, if you want to update the repository information, you would only need to supply the profile name and the block of options that affect the repository.

Usage format for smo profile update:

```
smo profile update
-profile profile_name
{-repository -dbname repo_dbname -host repo_host -login
-password repo_password -username repo_username -port repo_port}
{-database -dbname dbname -host dbhost -login -password db_password
-username db_username -port db_port [-sid db_sid]}
{-osaccount account_name}
[-rman {-tnsname tnsname -login -password rman_password -username
rman_username} | -controlfile ]
[-asm -host asm_host -login -password asm_password -username
asm_username -port asm_port -sid asm_sid ]
[-retain number]
[-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the smo profile update command.

Note

To create a profile, you use the command `smo profile create` instead of `smo profile update`.

Requirement	Argument
Specify the name of the profile. This name can be up to 30 characters long and must be unique for that host.	
◆ <code>-profile <i>profile_name</i></code>	<i>name of the profile</i>
Enter the name of the database. Use either the global name or the SID. You must also specify the connection information for the repository. This is the information that SnapManager needs in order to connect to the Oracle database. You must supply: ◆ The username and password for connecting to the Oracle database containing the repository. ◆ The port number to use to connect.	
◆ <code>-repository -dbname <i>db_name</i></code>	<i>name of the database</i>
◆ <code>-login</code>	~
❖ <code>-username <i>username</i></code>	<i>username for connecting to the database where the repository resides</i>
❖ <code>-password <i>password</i></code>	<i>password for connecting to the database where the repository resides</i>
◆ <code>-host <i>host</i></code>	<i>name or IP address of the database host</i>
◆ <code>-port <i>port_number</i></code>	<i>port number for connecting to the database</i>
Specify the database to be operated on and the information SnapManager needs in order to connect to the database: ◆ The name of the database you want in this profile. ◆ The username and password for connecting to the Oracle database. ◆ The database host (either the host name or its IP address). ◆ The port number to use to connect. ◆ In some cases, you must also include the SID. By default, SnapManager uses the database name as the SID. If the SID is different from the database name, you must specify it with the <code>-sid</code> argument. For instance, if you are using RAC, you must use this argument to specify the SID of the RAC instance on the RAC node from which SnapManager is executed.	
◆ <code>-dbname <i>db_name</i></code>	<i>name of the database being profiled</i>
◆ <code>-host <i>host</i></code>	<i>name of the database host</i>
◆ <code>-login</code>	~

Requirement	Argument
❖ -username <i>username</i>	<i>username for connecting to the database</i>
❖ -password <i>password</i>	<i>password for connecting to the database</i>
◆ -port <i>port_number</i>	<i>port number for connecting to the database</i>
◆ -sid <i>sid</i>	<i>SID for connecting to the database</i>
(Optional) If you want to catalog the SnapManager backups with RMAN, include the RMAN connection information in the profile also. To use RMAN, you must supply: ◆ The option <code>-rman</code>. ◆ (Optional) The <code>-controlfile</code> option if you want to use the control files instead of the RMAN catalog. ◆ If you do not use the <code>-controlfile</code> option, you must provide: ❖ The username and password for connecting to the RMAN catalog. ❖ The tnsname connection name (this is defined in the <code>tnsname.ora</code> file).	
◆ -rman	~
❖ -controlfile	~
◆ -login	~
❖ -username <i>username</i>	<i>username for connecting to the RMAN catalog</i>
❖ -password <i>password</i>	<i>password for connecting to the RMAN catalog</i>
◆ -tnsname <i>tnsname</i>	<i>connection name in the tnsname file</i>
Specify the name of the Oracle database user account. This is the account that SnapManager uses to perform the Oracle operations such as startup and shutdown. It is typically the user who owns the Oracle software on the host; for example, oracle.	
◆ -osaccount <i>owner_name</i>	<i>name of the Oracle database owner account</i>

Requirement	Argument
(Optional) If you use the Oracle Automatic Storage Management (ASM) instance, you must specify the information SnapManager needs in order to connect to it: ◆ The username and password for connecting to the Oracle database. ◆ The port number to use to connect. ◆ In some cases, you must also include the SID. By default, SnapManager uses the database name as the SID. If the SID is different from the database name, you must specify it with the <code>-sid</code> argument. For instance, if you are using RAC, you must use this argument to specify the SID of the RAC instance on the RAC node from which SnapManager is executed.	
◆ <code>-asm name</code>	<i>name of the ASM instance</i>
◆ <code>-login</code>	<i>~</i>
❖ <code>-username username</code>	<i>username for connecting to ASM</i>
❖ <code>-password password</code>	<i>password for connecting to ASM</i>
◆ <code>-port port_number</code>	<i>port number for connecting to ASM</i>
◆ <code>-sid sid</code>	<i>SID for connecting to the database</i>
(Optional) This option applies only to successful backups that were not used to clone a database. It does not affect unsuccessful backups, which SnapManager keeps, or backups used to clone a database. Specify the number of successful backups you want to SnapManager to retain. Enter an integer. If you do not supply a number here, SnapManager replaces the previous backup each time it creates a new backup. Note_____ You can specify any integer you like here; SnapManager does not have a limit. If you keep every backup, you may eventually reach the snapshot limit or run out of space. _____ When you include the retain option, SnapManager keeps the most recent successful backups up to the number you specify. When SnapManager produces a successful backup that causes it to exceed this number, it deletes the oldest successful backup, unless that backup was used to clone a database. SnapManager does not delete backups used to create clones or unsuccessful backups. Use the <code>sno backup delete</code> operation to remove those backups. Note_____ The unsuccessful backups do not affect the storage. Only the successful backups require space on the storage. _____	

Requirement	Argument
◆ <code>-retain <i>number</i></code>	<i>number of backups that are kept</i>
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “ Common options for handling messages shown on the console ” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Example of smo profile update command line

The following is a sample command line that uses the `smo profile` command.

```
# smo profile update -profile SMOKE_NFSTEST_BISMARCK -database -
dbname NFSTEST -sid NFSTEST -login -username sys -password oracle -
port 1521 -host bismarck
```

Output:

```
Operation Id [8abc01ec0e78ec33010e78ec3b410001] succeeded.
```

The smo profile verify command

Using the smo profile verify command

The `smo profile verify` command lets you confirm the profile set up.

Usage format for smo check:

```
smo profile verify
  -profile profile_name
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo profile verify` command.

Requirement	Argument
Enter the name of profile you created previously. SnapManager checks the environment for this profile. This profile defines all the credentials that SnapManager requires when working with a database. As part of the verification process, the <code>smo profile verify</code> command confirms that the database included in this profile is accessible.	
◆ <code>-profile <i>profile_name</i></code>	<i>name of an existing profile</i>
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo profile verify command line

The following is a sample command line that uses the `smo profile verify` command.

```
# smo profile verify -profile SMOKE_NFSTEST_BISMARCK
```

Output:

```
Operation Id [Nd7169f567e8a4c6c9700b1ebac49993b] succeeded.
```

The smo repository create command

Using the smo repository create command

The smo repository create command lets you create repositories in which to store database profiles and associated credentials.

Usage format for smo repository create:

```
smo repository create
    {-repository -dbname db_name -login -username username -password
    password -port port_number -host host}
    [-sid sid]
    [-noprompt]
    [-force]
```

The following is a summary of the information you need to supply when you use the smo repository create command.

Requirement	Argument	
Specify the name of the database.		
◆ -repository -dbname <i>db_name</i>	<i>name of the database the to be associated with the repository.</i>	
Specify the host name.		
◆ -host <i>host_name</i>	<i>name of the host</i>	
Specify the login username.		
◆ -login	~	
❖ -username <i>username</i>	<i>username for login</i>	
❖ -password <i>password</i>	<i>password for login username</i>	
◆ -port <i>port_number</i>	<i>port number for connecting to the database</i>	
◆ -sid <i>sid</i>	<i>system id number of the database</i>	
(Optional) Decide if you want SnapManager to attempt to force the creation of the repository.		
◆ -force	~	

Requirement	Argument	
(Optional) Decide if SnapManager should issue a prompt before creating the repository database if you also use the <code>-force</code> option. Using the <code>-noprompt</code> option insures the prompt does not appear, making it easier to create repositories using a script.		
◆ <code>-noprompt</code>		
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “ Common options for handling messages shown on the console ” on page 136.		
◆ <code>-quiet</code>	~	
◆ <code>-verbose</code>	~	

Example of smo repository create command line

The following are sample command lines that use the `smo repository create` command.

```
# smo repository create -repository -dbname SMOREPO -host bismarck
-login -username bismarck_smoke_user -password netapp -port 1521
-force
```

Output:

```
SMO-09207 [WARN ]: Repository schema with version 13 exists as
bismarck_smoke_user on jdbc:oracle:thin:@//bismarck:1521/SMOREPO.
```

The smo repository delete command

Using the smo repository delete command

The `smo repository delete` command lets you delete repositories in which to store database profiles and associated credentials.

Usage format for smo repository delete:

```
smo repository delete
    {-repository -dbname db_name -login -username username -password
    password -port port_number -host host}
    [-noprompt]
    [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo repository delete` command.

Requirement	Argument	
Specify the name of the database.		
◆ -repository -dbname <i>db_name</i>	<i>name of the database the to be associated with the repository.</i>	
Specify the host name.		
◆ -host <i>host_name</i>	<i>name of the host</i>	
Specify the login username.		
◆ -login	~	
❖ -username <i>username</i>	<i>username for login</i>	
❖ -password <i>password</i>	<i>password for login username</i>	
◆ -port <i>port_number</i>	<i>port number for connecting to the database</i>	
◆ -sid <i>sid</i>	<i>system id number of the database</i>	
(Optional) Decide if SnapManager should issue a prompt before deleting the repository database warning you about not being able to access the repository or the associated profiles, backups or clones after it is deleted. Using -noprompt insures the prompt does not appear, making it easier to delete repositories using a script.		
◆ -noprompt		

Requirement	Argument	
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “ Common options for handling messages shown on the console ” on page 136.		
◆ -quiet	~	
◆ -verbose	~	

Example of smo repository delete command line

The following are sample command lines that use the `smo repository delete` command.

```
# smo repository delete -repository -dbname repo_dbname -host
repo_host -login -username user1 -password password1 -port 1521
```

The smo repository update command

Using the smo repository update command

The `smo repository update` command lets you update repositories in which to store database profiles and associated credentials when you upgrade SnapManager for Oracle.

Attention

It is highly recommended that you backup existing repository databases prior to initiating the update process. A failure during the update process could leave your repository database in an inconsistent state.

Usage format for smo repository update:

```
smo repository update
  {-repository -dbname db_name -login -username username -password
  password -port port_number -host host}
  [-noprompt]
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo repository update` command.

Requirement	Argument	
Specify the name of the database.		
◆ -repository -dbname <i>db_name</i>	<i>name of the database the to be associated with the repository.</i>	
Specify the host name.		
◆ -host <i>host_name</i>	<i>name of the host</i>	
Specify the login username.		
◆ -login	~	
❖ -username <i>username</i>	<i>username for login</i>	
❖ -password <i>password</i>	<i>password for login username</i>	
◆ -port <i>port_number</i>	<i>port number for connecting to the database</i>	
◆ -sid <i>sid</i>	<i>system id number of the database</i>	

Requirement	Argument	
(Optional) Decide if SnapManager should issue a prompt recommending you backup the repository database before updating it. Using the <code>-noprompt</code> option insures the prompt does not appear, making it easier to update repositories using a script.		
◆ <code>-noprompt</code>		
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “ Common options for handling messages shown on the console ” on page 136.		
◆ <code>-quiet</code>	~	
◆ <code>-verbose</code>	~	

Example of sm repository update command line

The following are sample command lines that use the `sm repository update` command.

```
# sm repository update -repository -dbname repo_dbname -host
repo_host -login -username user1 -password password1 -port 1521
```

The smo system dump command

Using the smo system dump command

The `smo system dump` command lets you create a dump file containing information about the environment. This file can be helpful if you need to troubleshoot a problem with the operation.

Usage format for smo system dump:

```
smo system dump
  [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo profile dump` command.

Requirement	Argument
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ -quiet	~
◆ -verbose	~

Examples of smo system dump command line

The following is a sample command line that uses the `smo profile dump` command.

```
# smo system dump
```

Output:

```
SMO-15034: Output for the dump command is stored in file
"/tmp/smo_dump_results.jar".
Dump file created. Host: bismarck Path: /tmp/smo_dump_results.jar
```

The smo system verify command

Using the smo system verify command

The `smo system verify` command lets you confirm that all the components of the environment required to run SnapManager are set up correctly.

Usage format for smo check:

```
smo system verify
    [-quiet | -verbose]
```

The following is a summary of the information you need to supply when you use the `smo system verify` command.

Requirement	Argument
(Optional) Decide how you want messages displayed. For details on the options to control message output, see “Common options for handling messages shown on the console” on page 136.	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo system verify command line

The following is a sample command line that uses the `smo system verify` command.

```
# smo system verify
```

Output:

```
SMO-13505 [INFO ]: Snapdrive verify passed.
SMO-13037 [INFO ]: Successfully completed operation: System Verify
SMO-13049 [INFO ]: Elapsed Time: 0:00:00.559
Operation Id [N4f4e910004b36cfecee74c710de02e44] succeeded.
```

The smo version command

Using the smo version command

The smo version command lets you determine which version of SnapManager you are running on your local host. If you include the `-verbose` option with this command, it also displays the build date.

Usage format for smo version:

```
smo version [-verbose]
```

The following is a summary of the information you can supply when you use the `smo version` command.

Requirement	Argument	
(Optional) Decide if you want messages from the CLI displayed on the console. If you include the <code>-verbose</code> option, SnapManager displays the build date and the contents of each profile. If you omit the this option, SnapManager only displays the version number and the profile name.		
◆ <code>-verbose</code>	<code>~</code>	

Example of smo version command line

The following is a sample command line that uses the `smo verify` command.

```
# smo version
```

Output:

```
SnapManager for Oracle: Version 2.0
```

Index

A

- abort operation
 - executing from CLI 95
 - general steps 95
 - graceful cleanup 94
 - incomplete backup 94
 - interrupting executing operation 94
- ASM
 - backup fails with raw LUNs on qtree 125
 - confirming it is accessible 62
 - Linux limitation 31
- ASM, RAC, and RMAN integration 5
- auto recovery
 - defined 7

B

- backup 23
 - cloning 78
 - defined 7
 - display profile information 173
 - examples of retaining 23
 - labels 134
 - problem seeing in RMAN 124
 - removing 101
 - repository 12
 - retaining
 - retaining backups 23
 - sliding window 23
 - steps to display information 100
 - unable to display in GUI 121
 - verifying 71
- backup create operation
 - CLI command 137
 - CLI example 141
 - command parameters 137
- backup delete operation
 - CLI example 143
- backup list operation
 - CLI command 144
 - CLI example 144
 - command parameters 144
- backup mount command
 - backup mount 146
- backup mount operation
 - CLI command 146
 - CLI example 147
 - command parameters 146
- backup operation
 - algorithm for full backup 20
 - algorithm for partial backup 21
 - executing from CLI 69
 - fails without message 126
 - files included 14
 - full 19
 - hangs 125
 - incomplete 94
 - information required 68
 - offline backup 68
 - online backup 68
 - partial 19
 - types of backups 14
 - verifying 59
 - viewing files 14
- backup restore operation
 - CLI command 148
 - CLI example 150
 - command parameters 148
- backup unmount command
 - backup unmount 151
- backup unmount operation
 - CLI command 151
 - CLI example 152
 - command parameters 151
- backup verify operation
 - CLI command 153
 - CLI example 154
 - command parameters 153
 - executing from CLI 71
 - steps 71
- backups
 - displaying information about 100
 - removing 101
- before issuing commands

start server 49

C

case-sensitive

command options 135

changing

existing profiles 98

profile properties 98

CLI

backup list command 144

backup mount 146

backup restore command 148

backup unmount command 151

backup verify command 153

clone create operation 155

clone template operation 157

cmdfile 158

delete backup command 142

dump command 118

dump, system 188

executing abort operation 95

executing backup operation 69

executing backup verify operation 71

executing clone operation 89

executing list operation 100

executing mount operation 104

executing remove operation 102

executing restore operation 74

executing unmount operation 104

help command 160

modifying profile 98

operation abort command 161

operation dump command 163

profile create command 54, 64, 165

profile delete command 99, 170

profile dump command 172

profile list command 173

profile show command 174

profile update command 176

starting SnapManager 34

system dump command 188

version command 190

clone create operation

CLI command 155

CLI example 156

command parameters 155

clone operation

executing from CLI 89

fails with NID-00100 122

general steps 88

problem with mount point 122

use unique SID 122

Clone Specification 80

clone specification file

creating 80

clone template operation

CLI command 157

CLI example 157

command parameters 157

cloning backups 78

clone specification file 80

cloning RAC database 91

creating Oracle files 79

different SIDs 79

files created 79

RAC database 79

setting up RAC clone 92

cmdfile

command parameters 158

cmdfile command

CLI example 159

cmdfile operation

CLI command 158

command

backup list 144

operation abort 161

command options

case-sensitive 135

displaying messages 136

quiet 136

verbose 136

commands

backup create 137

backup restore 148

backup verify 153

clone create 155

clone template 157

cmdfile 158

general information 134

- help 160
- manage list 144
- manage mount 146
- manage unmount 151
- operation abort 161
- operation dump 163
- Oracle dbv 59, 71
- profile create 165
- profile delete 170
- profile dump 172
- profile list 173
- profile show 174
- profile update 176
- profile verify 181
- recover option 148
- system dump 188
- system verify 189
- version 190
- Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products 131
- Components 80
- controlfile
 - recovery slow 124
- controlfiles
 - included in backup 14
- conventions
 - formatting ix
 - keyboard x
- creating
 - profiles 61

D

- Data ONTAP Block Access Management Guide 132
- database
 - confirming it is accessible 62
 - recovering 73
 - restoring 73
- databases
 - cloning 79
 - locked during operations 11
 - RAC 79
- DBA
 - using SnapManager 2

- delete
 - profile 99
 - repository 97
- delete backup operation
 - CLI command 142
- deleting
 - profile 99
 - profiles 99
 - repositories 97
- Documentation
 - NOW Description Page 131
- documentation
 - additional 131
 - Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products 131
 - Data ONTAP Block Access Management Guide 132
 - FCP host attach kit 132
 - SnapDrive for UNIX Quick Start 132
 - SnapManager Compatibility Matrix 131
 - SnapManager Release Notes 131
 - System Configuration Guide 132
- dump command
 - CLI command 118
 - executing 119
 - information producted 118
- dump file
 - creating 119
 - locating 118

E

- environment
 - verifying 61, 62
- environment variables
 - checking 59
 - verifying
 - checking 62
- error messages 136
 - code ranges 129
- examples
 - backup create command 141
 - backup delete command 143
 - backup list command 144

- backup mount command 147
- backup restore command 150
- backup unmount command 152
- backup verify command 154
- clone create command 156
- clone template command 157
- cmdfile command 159
- help command 160
- operation abort command 162
- operation dump command 163
- profile create command 168
- profile delete command 171
- profile dump command 172
- profile list command 173
- profile show command 174
- profile update command 180
- profile verify command 181
- recover option 150
- repository create command 183
- repository create option 183
- repository delete command 185
- repository delete option 185
- repository update command 187
- repository update option 187
- retaining backups 23
- system dump command 188
- system verify command 189
- version command 190

F

- file system
 - full 126
- full backup
 - algorithm 20
 - compared to partial 19
 - defined 7

G

- GUI
 - installing 47
 - starting SnapManager 34
 - unable to display backups 121
- GUID
 - defined 7

- view using list 134

H

- help command
 - CLI example 160
 - command parameters 160
- hosts
 - installation steps on Linux host 46

I

- in-band events 26
- informational messages 136
- installing
 - GUI 47
 - steps for Linux host 46

L

- label option
 - case sensitive 134
- licenses
 - required 42
- Linux host
 - installation steps for SnapManager 46
 - requires ASMLib 31
 - uninstalling SnapManager 107
- list operation
 - executing from CLI 100
 - general steps 100
- lock
 - during operations 11
- log files
 - file system full 126
 - included in backup 14

M

- management
 - tasks 93
- managing
 - SnapManager operations 93
- mount operation
 - executing from CLI 104
 - general steps 103

- mount point
 - problem with clone 122

N

- NID-00100 error 122
- NOW site
 - downloading software 44

O

- OCFS
 - unsupported 31
- offline
 - defined 7
- offline backup 68
- online
 - defined 7
- online backup 68
- operation abort
 - CLI command 161
 - CLI example 162
 - command parameters 161
- operation dump command
 - CLI command 163
 - CLI example 163
 - parameters 163
- operational states
 - failed 25
 - in progress 25
 - in-band (recoverable) 26
 - incomplete 25
 - out-of-band (unrecoverable) 26
 - SnapManager 25
 - succeeded 25
- operations
 - managing SnapManager 93
- out-of-band
 - events 26
- Overview, ASM, RAC, and RMAN integration 5
- Overview, repositories and profiles 6

P

- partial backup
 - algorithm 21

- compared to full 19
 - defined 7
- persistent state
 - maintained by SnapManager 94
- PIT recovery
 - defined 7
- profile
 - changing existing 98
 - create 61
 - creating dump files 172
 - creating from CLI 54, 64
 - defined 7
 - delete 99
 - deleting from CLI 99
 - explained 10
 - information needed to create 63
 - modifying from CLI 98
 - verifying 61
- profile create operation
 - CLI command 165
 - CLI example 168
 - command parameters 165
- profile delete operation
 - CLI command 170
 - command parameters 170
 - example 171
- profile dump command
 - CLI command 172
 - CLI example 172
 - parameters 172
- profile list operation
 - CLI command 173
 - CLI example 173
 - command parameters 173
- profile show operation
 - CLI command 174
 - CLI example 174
 - command parameters 174
- profile update
 - command parameters 176
- profile update operation
 - CLI command 176
 - CLI example 180
- profile verify operation
 - CLI command 181

- CLI example 181
- command parameters 181
- profiles
 - changing existing 98
- properties
 - profile 98
- protocol
 - license required 42
- protocols
 - accepted 29, 30

Q

- qtrees
 - ASM issue with raw LUNs 125
- quiet command option
 - using 136

R

- RAC database
 - clone options 91
 - cloning 79, 91
 - setting up clone 92
 - specify one node 11
- recover
 - defined 7
 - option
 - general steps 74
- recover option
 - auto 7
 - command parameters 148
 - example 150
 - point-in-time 7
 - recovering data 73
 - used with restore 73
 - using with restore command 74
- recoverable events 26
- recovery operations
 - types 14
- remove operation
 - executing from CLI 102
 - general steps 101
- repositories
 - creating 51
 - deleting 97

- multiple 12
- updating 51
- repository
 - backing up 12
 - create 51
 - defined 7
 - delete 97
 - delete repositories
 - deleting 97
 - explained 10
 - information needed to create 52
 - update 51
- repository create operation
 - CLI example 183
- repository create option
 - example 183
- repository delete operation
 - CLI example 185
- repository delete option
 - example 185
- repository update operation
 - CLI example 187
- repository update option
 - example 187
- restore
 - defined 7
- restore operation
 - executing from CLI 74
 - general steps 74
 - options 73
 - types 14
- retaining backups
 - examples 23
- RMAN
 - confirming it runs 62
 - problem seeing backups 124

S

- server
 - starting 49
- shell
 - using cmdfile to enter commands 134
- SID
 - duplicate fails 122

SIDs

- used with clones 79

- sliding window 23

- SnapDrive for UNIX

- required 29

- SnapManager

- additional documentation 131

- backing up data 68

- checking the version 106

- downloading software 44

- getting 44

- installation steps on Linux 46

- installing GUI 47

- licenses 42

- limitations 31

- operational states 25

- starting from CLI 34

- starting from GUI 34

- uninstalling from Linux host 107

- uninstalling the GUI 107

- working with RAC 11

- SnapManager Compatibility Matrix 131

- SnapManager Release Notes 131

- SnapRestore

- required 42

- snapshot

- defined 7

- software

- downloading 44

- SQLPlus

- confirming it runs 59, 62

- starting

- the server 49

- storage manager

- using SnapManager 2

- System Configuration Guide 132

- system dump command

- CLI command 188

- CLI example 188

- parameters 188

- system verify operation

- CLI command 189

- CLI example 189

- command parameters 189

T

- tasks

- managing SnapManager 93

- terms

- auto recovery 7

- backup 7

- full backup 7

- GUID 7

- offline 7

- online 7

- partial backup 7

- PIT recovery 7

- profile 7

- recover 7

- repository 7

- restore 7

- snapshot 7

- troubleshooting

- ASM backup fails 125

- backup fails without message 126

- backup hangs 125

- clone mount point issue 122

- clone needs unique SID 122

- controlfile recovery slow 124

- file system 126

- NID-00100 error 122

- problem seeing RMAN backups 124

- unable to display GUI backups 121

- using the dump command 118

U

- Understanding repositories and profiles 6

- uninstalling

- SnapManager from Linux host 107

- SnapManager GUI 107

- unmount operation

- executing from CLI 104

- general steps 103

- unrecoverable events 26

V

- verbose command option

- using 136

- verify
 - backup 71
 - environment 61
- verifying
 - profile 61
- version
 - checking 106
- version command
 - CLI command 190

- CLI example 190

W

- warning messages 136
- Windows system
 - installing GUI 47
 - uninstalling SnapManager GUI 107