

SnapManager® 2.1 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 215-03035_A0
April 2007

Copyright and trademark information

Copyright information

Copyright © 1994–2007 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, FAServer, FilerView, FlexVol, Manage ONTAP, MultiStore, NearStore, NetCache, SecureShare, SnapDrive, SnapLock, SnapManager, SnapMirror, SnapMover, SnapRestore, SnapValidator, SnapVault, Spinnaker Networks, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, SyncMirror, Topio, VFM, and WAFL are registered trademarks of Network Appliance, Inc. in the U.S.A. and/or other countries. Cryptainer, Cryptoshred, Datafort, and Decru are registered trademarks, and Lifetime Key Management and OpenKey are trademarks, of Decru, a Network Appliance, Inc. company, 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, EdgeFiler, FlexClone, FlexShare, FPolicy, HyperSAN, InfoFabric, LockVault, NOW, NOW NetApp on the Web, ONTAPI, 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	xi
Chapter 1	SnapManager for Oracle Overview	1
	What SnapManager for Oracle Can Do for You	2
	SnapManager Glossary	5
Chapter 2	SnapManager Concepts	7
	Basic SnapManager Concepts	8
	Working with repositories	10
	Approaches to backing up, restoring, and recovering data	12
	How SnapManager handles backups	17
	SnapManager’s policy for retaining backups.	21
	SnapManager operation states	23
	SnapManager security	25
Chapter 3	SnapManager Deployment Considerations	27
	SnapManager for Oracle requirements	28
	Prerequisites and considerations	29
	Limitations and Restrictions	32
Chapter 4	SnapManager Interfaces	35
	Interface choices	36
	Using the SnapManager for Oracle CLI	37
	Using the SnapManager for Oracle GUI	38
Chapter 5	Installing or Upgrading SnapManager	45
	Installing and upgrading SnapManager tasks.	46
	Downloading the SnapManager software package.	49

	Installing or upgrading the SnapManager software packages	51
Chapter 6	Configuring SnapManager	55
	Starting up the CLI or GUI	56
	Allowing access to SnapManager secured resources.	57
	Updating and creating repositories	60
	Verifying, starting and stopping the server.	64
Chapter 7	Quick Start Example Procedure	65
	Before you begin	66
	Example procedure	68
Chapter 8	Creating Profiles.	73
	Verifying the environment	74
	Creating a profile	75
	Verifying the profile	79
Chapter 9	Backing Up and Restoring Databases	81
	Creating database backups	82
	Verifying a backup	85
	Restoring and recovering a database backup	87
Chapter 10	Cloning Databases Using SnapManager.	91
	General SnapManager database cloning information	92
	Creating a clone specification	95
	Cloning a database from a backup	103
	Cloning a database in its current state	105
	Cloning a database to an alternate host.	106
	Cloning with RAC databases	108

Chapter 11	Managing SnapManager Operations	.111
	Managing Credentials	.112
	Aborting an operation.	.114
	Issuing SnapManager CLI Commands from an alternate host	.117
	Deleting repositories	.118
	Changing an existing profile	.119
	Deleting profiles	.120
	Listing backups	.121
	Showing backup details.	.123
	Updating the backup retention policy	.127
	Deleting backups	.129
	Mounting and unmounting backups	.131
	Listing clones	.134
	Showing clone details.	.135
	Deleting clones	.137
	Checking the version of SnapManager	.138
	Uninstalling SnapManager software	.139
 Chapter 12	 Advanced Topics	 .141
	Performing a block level restore using RMAN.	.142
 Chapter 13	 Troubleshooting	 .147
	Questions to consider.	.148
	Gathering troubleshooting information.	.150
	Troubleshooting upgrade issues	.153
	General troubleshooting tips for using the GUI	.154
	General troubleshooting tips for working with clones	.156
	General troubleshooting tips for SnapManager	.157
	Common error conditions.	.171
	Where to go for more information	.173

SnapManager Command Reference	.175
Collecting information needed by SnapManager Commands	.177
Conventions for entering names and command options	.178
Common options for handling messages shown on the console	.179
The smo backup create command	.180
The smo backup delete command	.184
The smo backup list command	.186
The smo backup mount command	.188
The smo backup restore command	.190
The smo backup show command	.193
The smo backup update command	.196
The smo backup unmount command	.198
The smo backup verify command	.200
The smo clone create command	.202
The smo clone delete command	.204
The smo clone list command	.205
The smo clone show command	.206
The smo clone template command	.208
The smo cmdfile command	.210
The smo credential clear command	.212
The smo credential list command	.213
The smo credential set command	.214
The smo help command	.216
The smo operation abort command	.217
The smo operation dump command	.219
The smo profile create command	.221
The smo profile delete command	.226
The smo profile dump command	.227
The smo profile list command	.228
The smo profile show command	.229

The smo profile sync command	.231
The smo profile update command	.233
The smo profile verify command.	.237
The smo repository create command	.238
The smo repository delete command	.240
The smo repository show command	.242
The smo repository update command	.244
The smo_server restart command	.246
The smo_server start command.	.247
The smo_server status command	.248
The smo_server stop command.	.249
The smo system dump command.	.250
The smo system verify command.	.251
The smo version command	.252
Index	.253

Preface

- About this guide

This document describes how to install, configure, and use SnapManager® 2.1 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 for 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 or Solaris host where the 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
- ◆ [“SnapManager Glossary”](#) on page 5

What SnapManager for Oracle Can Do for You

SnapManager from a DBA's perspective

SnapManager for Oracle simplifies the backup of data in Oracle databases for Database Administrators (DBA). SnapManager:

- ◆ 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 of a database.
- ◆ Reduces the mean time to recover a database by using SnapRestore.
- ◆ Works with Oracle products to enable DBAs to continue using their current tool sets.

SnapManager from a storage administrator's perspective

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

- ◆ 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).

Products SnapManager works with

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.

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.

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.

The advantage to this is that 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, 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 (Logical Units). You can also use SnapManager to back up a database when there are multiple databases in the same volume or LUN.

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, 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 it has backed up. The clone is separate from the actual backup, so you can modify it without changing the backup source. The clone can be located on the same host as the database, or on a different host as long as it meets the requirements detailed in “[Cloning a database to an alternate host](#)” on page 106. SnapManager uses the FlexClone feature, which is part of Data ONTAP, to create the clone. FlexClone enables SnapManager to use snapshots of the database 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

SnapManager also works with and provides an integration point for such Oracle capabilities as Oracle's Automatic Storage Management (ASM) feature and Oracle's Recovery Manager (RMAN).

SnapManager can catalog the backups it makes with RMAN. If using RMAN, a DBA can make use of SnapManager's backups and preserve the value of all RMAN functions, such as block level restore. The RMAN recovery catalog should not be in the same database as the database that is being backed up.

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.

SnapManager also integrates with Real Application Clusters (RAC). Invoke SnapManager from a RAC database node to create the backup, restore and recover the database, and clone the database.

SnapManager Glossary

Introduction

The following are terms you should be familiar with as you read this document.

Terms

Alternate host: A host other than the host on which the database resides.

Authentication: The SnapManager process by which users gain access to a host or other secured resource.

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

Credentials: The passwords that allows users to access secured resources.

Full backup: A backup containing a snapshot copy of the entire database, including data files, control files, and archive logs.

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: A backup containing a snapshot 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. 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. 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.

Secured resource: For SnapManager, those resources requiring authentication for access, such as hosts, repositories and profiles.

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

Source database: When cloning, the database that is being cloned.

Source host: When cloning to an alternate host, the host on which the original database resides.

Target database: When cloning, the new database that is being created.

Target host: When cloning to an alternate host, the host on which the new database will reside.

About this chapter

This chapter provides information about SnapManager concepts.

This chapter includes the following topics:

- ◆ “[Basic SnapManager Concepts](#)” on page 8
- ◆ “[Working with repositories](#)” on page 10
- ◆ “[Approaches to backing up, restoring, and recovering data](#)” on page 12
- ◆ “[How SnapManager handles backups](#)” on page 17
- ◆ “[SnapManager’s policy for retaining backups](#)” on page 21
- ◆ “[SnapManager operation states](#)” on page 23
- ◆ “[SnapManager security](#)” on page 25

Basic SnapManager Concepts

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. Repositories hold data about the operations performed on the profiles.

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

The repository contains information about the profile and its backups and clones, 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 being backed up or a different storage 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.

Understanding profiles

The profile contains the information required for creating the backup and for performing other SnapManager operations, such as backups and clones. It also contains information on the operations that have been performed, including:

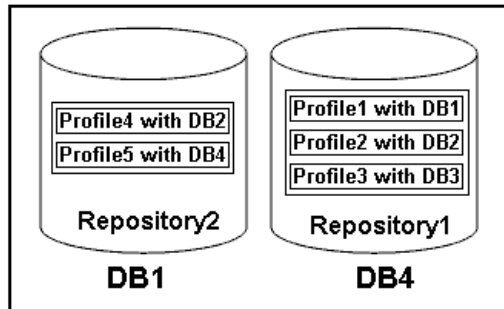
- ◆ Database credentials
- ◆ How many backups to retain

The credentials stored include the username/password pairs for accessing the host, 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 reference only one database. That same database can be referenced by more than one profile. You cannot access a backup created using one profile from a different profile, even if both profiles are associated with the same database.

SnapManager places a lock on the database during an operation to prevent two incompatible operations from being performed simultaneously.

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

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 DB4, a backup of it is made each time you create a backup of DB4. Conversely, a backup is created of Repository2 each time you create a backup of DB1. 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 DB4, and Repository2 is located in the database DB1.

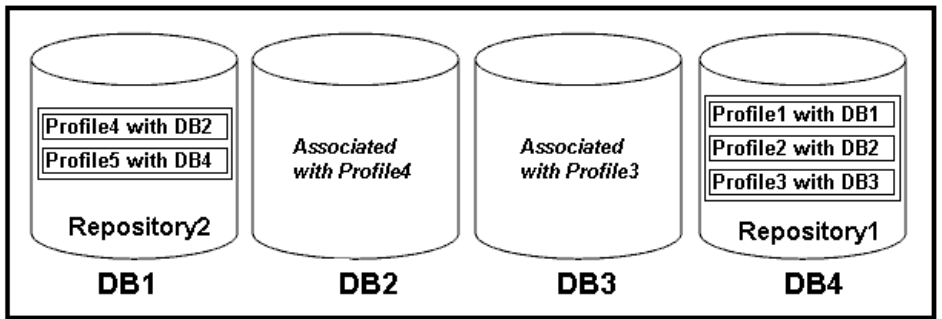
Repository1 has three profiles:

- ◆ Profile1, which is associated with database DB1.
- ◆ Profile2, which is associated with DB2.
- ◆ Profile3, which is associated with DB3.

Repository2 has two profiles:

- ◆ Profile4, which is associated with database DB2.
- ◆ Profile5, which is associated with DB4.

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



The repositories are only in two of the databases. To create a backup of each repository, backup a profile associated with the database that contains the repository. For example, because Repository1 is in database DB4, create a backup of Repository1 by backing up Profile5, which is associated with DB4.

Repository2 is in database DB1, so create a backup of it by backing up Profile1, which is associated with that database.

Both repositories contain a profile associated with database DB2, so neither repository could be located in database DB2. SnapManager does not allow the 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 or portions of it. SnapManager offers the following choices:

- ◆ Backup operations. Each backup consists of one of the following:
 - ❖ The entire database
 - ❖ 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.

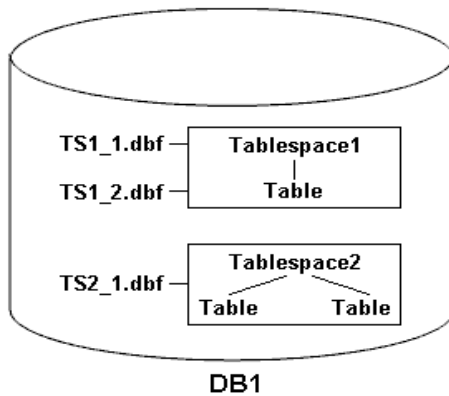
- ◆ Recovery operations. Use SnapManager to recover the restored data by specifying that it recover the data based on either a point in time or all available logs (i.e., the last transaction committed to the database). The point in time can be an Oracle SCN or a date and time (`yyyy-mm-dd:HH:MM:SS`).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

The following table summarizes the options available with the SnapManager backup, restore, and recover operations.

Backup	Restore	Recover using SnapManager
Entire database 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	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. There are benefits and consequences to each type. Base your decision on what is most important in your environment.

- ◆ 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 18 and “[Algorithm for performing a partial backup](#)” on page 19.

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 files.
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 files.
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 21 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. You also have the option to exclude a backup from the retention policy. The retention policy is engaged every time you take a new backup.

This number of backups retained is affected by the following rules:

- ◆ The retention policy applies only to successful backups. It does not apply to backups that are:
 - ❖ Mounted
 - ❖ Incomplete or failed (unsuccessful)
 - ❖ Designated as retain forever
 - ❖ Used to create a clone
- ◆ 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. An unsuccessful backup remains until you manually delete it.

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

- ◆ SnapManager keeps all backups used to create a clone.
- ◆ SnapManager always keeps the last backup taken.
- ◆ If you have designated the backup to be retained forever, the backup is not counted. If you have designated the backup to be retained normally and it is a successful backup, it is counted.

Set the retain option to have SnapManager keep a number of backups. Once the number of successful backups is reached, working backwards from the latest successful backup, SnapManager removes the oldest successful backup that is eligible for the retention policy.

Example of how SnapManager retains backups

The following table shows the backups taken, the status of the backup, and the SnapManager action based on a retention policy value set to retain three backups.

Backup Taken	Status	Retention Policy Action
Retention Policy Value set to 3		
5/5	Successful	Keep
4/27	Successful, Cloned	Skip
4/20	Successful, Mounted	Skip
4/13	Failed	Skip
4/6	Successful, designated to be retained forever	Skip
3/31	Successful	Keep
3/24	Successful	Keep
3/17	Successful	Delete

This keeps the total number of eligible successful backups to three. The total number of backups is now seven:

- ◆ 5/5, which was successful, is the first backup kept because it is the latest successful backup taken
- ◆ 4/27, which was used to make a clone, is skipped
- ◆ 4/20, which was mounted, is skipped
- ◆ 4/13, which failed, is skipped
- ◆ 4/6, which was designated as retain forever is skipped
- ◆ 3/31, which was successful, counts as the second successful backup kept
- ◆ 3/24, which was successful, counts as the third successful backup kept. This backup fulfills the retention policy and any other successful backup prior to that is deleted.
- ◆ 3/17, which was successful, is deleted

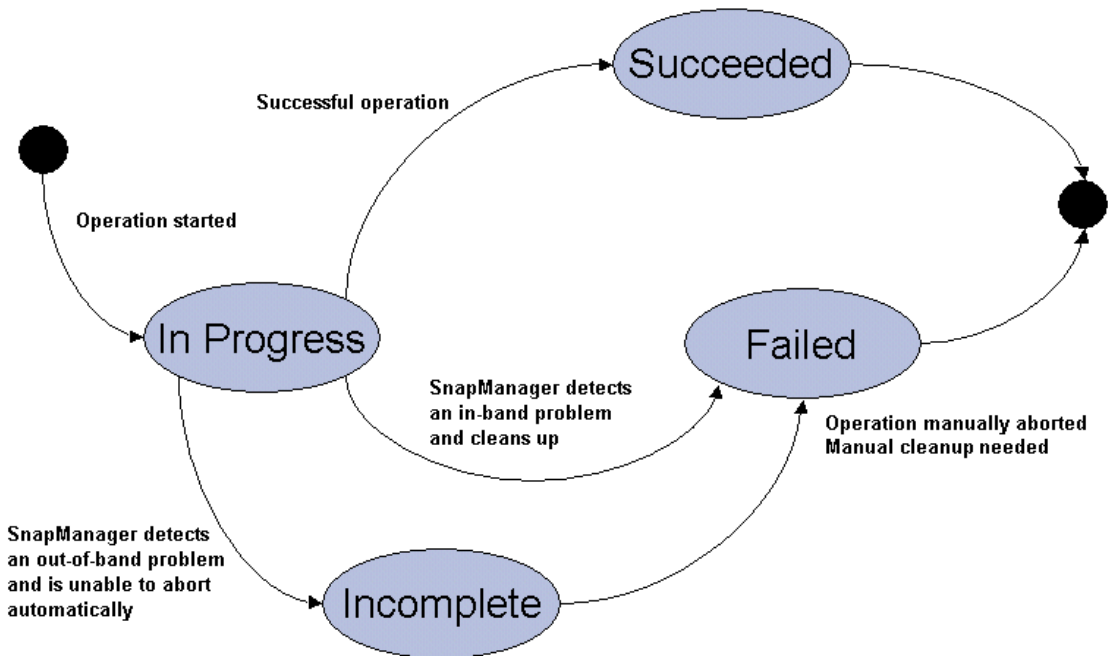
SnapManager operation states

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 event (i.e. out-of-band) occurred. In this case, SnapManager cannot perform its standard abort processing. You must manually abort the operation. Refer to the “[Aborting an operation](#)” on page 114 for more information.
- ◆ Failed. The operation failed. SnapManager automatically executed its abort process and cleaned 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.

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 abort the process. Refer to the [“Aborting an operation”](#) on page 114 for more information.

SnapManager security

About Secured Resources

SnapManager offers administrators the ability to manage access to certain resources, such as:

- ◆ Local and remote SnapManager hosts
- ◆ Repositories
- ◆ Profiles

As an administrator, you set the passwords for these resources. Users set their credentials to access only the resources on which they have permissions and need to execute operations.

Consider the case where User1 and User2 share a profile called Prof2.

Permission type	User1	User2
Host permissions	Host1, Host2	Host2, Host3
Repository permissions	Repo1	Repo1
Profile permissions	Prof1, Prof2	Prof2

In this case, User2 cannot backup Database1 in Host1 because he doesn't have permissions to access Host1. User1 cannot clone a database to Host3, because he doesn't have credentials to access Host3.

In the case where User1 and User2 do not have any shared profiles, assume User1 has permissions for hosts Host1 and Host2, and User2 has permissions for host Host2. User2 cannot even run non-profile commands (dump, system verify, etc.) on Host1.

About Oracle and Operating System authentication

There are two ways that SnapManager can interface with a database.

- ◆ Oracle authentication with a username, password, and port
- ◆ Operating System (OS) authentication with no username, password, or port

Note _____
OS Authentication is not supported for RAC databases.

GUI users have the opportunity to choose the authentication connection mode with which to access the database within the appropriate wizard screens. To authenticate via the OS, CLI users specify the OS Account and OS Group options on the command line.

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 and considerations](#)” on page 29
- ◆ “[Limitations and Restrictions](#)” on page 32

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.
- ◆ When using SnapManager for Oracle with ASM, you must use RMAN. Profiles for ASM databases require that RMAN be enabled.

Prerequisites and considerations

Prerequisites

Before using SnapManager:

- ◆ 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 for all hosts, storage systems and other components listed below.
- ◆ Set up the host system and storage systems. On the host, you must have:
 - ❖ A host operating system and appropriate patches installed.

Note

SAN Support on some platforms have specific minimum Oracle version requirements.

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 Utilities. 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 only if you mount, verify, and clone NFS database backups. It is also required if you register backups of NFS databases with 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	iSCSI, FCP, NFS
HP-UX	iSCSI, FCP, NFS

- ◆ If using NFS, ensure your file systems are mounted as recommended in Network Appliance™ Best Practice Guidelines for Oracle® at <http://www.netapp.com/library/tr/3369.pdf>.

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: Native JFS2
 - ❖ HP-UX: Native VxFS

Topologies

SnapManager supports the same host cluster and active/active storage system 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 controller 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.

Limitations and Restrictions

Oracle limitations

SnapManager 2.1 for Oracle has the following limitations with Oracle:

- ◆ There is no support for Oracle Cluster File System (OCFS).
- ◆ There may be certain version and patch limitations. Verify support with the SnapManager and SnapDrive Compatibility Matrix on the NOW site.

NFS limitations

SnapManager runs as root and must be able to access the file systems 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.
- ◆ If you are using iSCSI or FCP on Linux, use the Automatic Storage Management support library (ASMLib).
- ◆ Profiles for ASM databases must have RMAN enabled.
- ◆ ASM disks must be raw LUNs. Using files on an NFS or host-side file system as ASM disks is not supported.
- ◆ If you are not using OS authentication to connect to the ASM instance, you must manually register an ASM instance with the listener via `listener.ora`. In your `$ORACLE_HOME/network/admin/listener.ora` verify that the file contains the following information, and if necessary add it, substituting the path to `ORACLE_HOME` and the `SID` for your database:

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(ORACLE_HOME = /u01/app/oracle/10.1.0.2/product/10.1.0/db)
(SID_NAME = DBSID)
)
```

)

Now restart the both the listener and the ASM instance.

- ◆ ASM is not supported on IBM AIX or HP-UX.
- ◆ 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/library/tr/3329.pdf>) and TR 3349 (<http://www.netapp.com/library/tr/3349.pdf>). These reports provide information on how to lay out the LUNs for your ASM disk groups.

Other limitations and restrictions

- ◆ SnapManager cannot automatically archive the backups to, or restore archived backups from, another storage system. The snapshots exist on the same storage system as the primary data.
- ◆ SnapManager supports control files on a file system or in ASM and does not support control files on raw devices.
- ◆ Backups containing RAW tablespaces or ASM disks on Linux created with previous versions cannot be cloned using SnapManager 2.1 for Oracle and later.
- ◆ SnapManager supports databases on the V-Series storage systems, with the following requirements:
 - ❖ You must configure SnapDrive to set passwords for V-Series storage systems as is the current requirement with other storage systems.
 - ❖ SnapDrive for Unix cannot create a snapshot of a LUN or file residing in a qtree in a V-Series storage system if the underlying volume is not in the same V-Series storage system.
- ◆ SnapManager does not support reverting to a previous version of SnapManager.
- ◆ A database may not have files in more than one type of file system or volume manager. All files making up a database must reside on the same type of file system.

- ◆ SnapManager may be configured to catalog database backups with RMAN. If an RMAN recovery catalog is being used, the recovery catalog must be in a different database than the database that is being backed up.
- ◆ SnapDrive® for UNIX® supports more than one type of file system and/or volume manager on certain platforms. The file system and/or volume manager used for database files must be specified in the SnapDrive configuration file as the default file system/volume manager. Refer to the SnapDrive for UNIX 2.2 *Installation and Administration Guide* for more information about using this file.

Volume Management Restrictions

- ◆ You can have multiple disk groups for a database, however, all of these rules apply to all disk groups for a given database.
 - ❖ A disk group can contain one and only one file system.
 - ❖ Disk groups for the a database can be managed by only one volume manager.
- ◆ Raw devices and ASM disks are not supported with Volume Manager volumes.

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 36
- ◆ [“Using the SnapManager for Oracle CLI”](#) on page 37
- ◆ [“Using the SnapManager for Oracle GUI”](#) on page 38

Interface choices

Use SnapManager for Oracle in any 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 and from alternate hosts.
- ◆ 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.

The GUI has the following space requirements:

- ❖ Minimum 128 MB available disk space for the application
- ❖ Minimum 512 MB RAM available to run the application

Using the SnapManager for Oracle CLI

About the CLI

The CLI commands all start with `smo` (SnapManager for Oracle). For example, you have commands such as `smo repository create` or `smo backup create`.

Note

You cannot run the SnapManager CLI on the Windows host.

For a list of all the SnapManager CLI commands and an explanation of their options and arguments, see “[SnapManager Command Reference](#)” on page 175.

Using the SnapManager for Oracle GUI

About the GUI

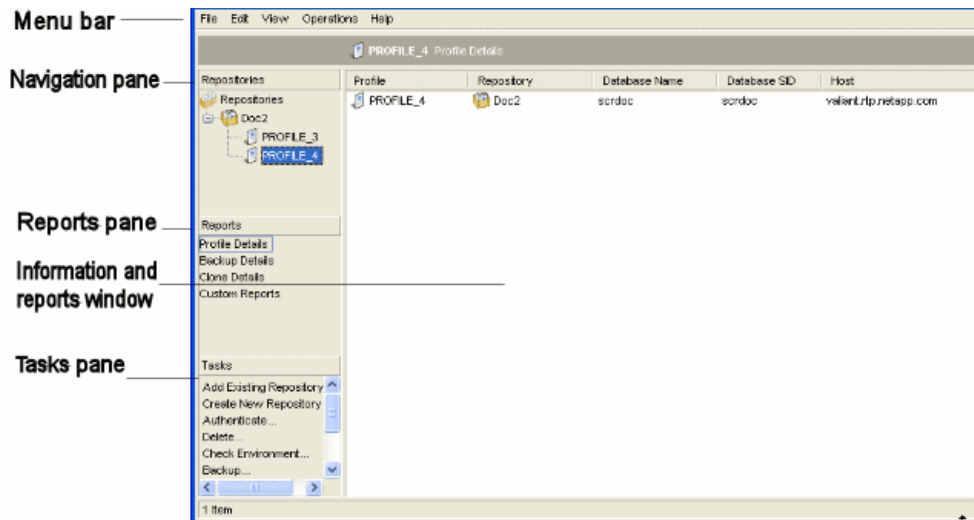
In addition to performing operations, the GUI provides a central method for managing a database's resources. There are several ways to perform SnapManager operations from the GUI.

- ◆ 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 components of 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.



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. Refer to "[How SnapManager handles backups](#)" on page 17 and "[SnapManager's policy for retaining backups](#)" on page 21 for more information.
- ◆ **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, backups and clones. 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.

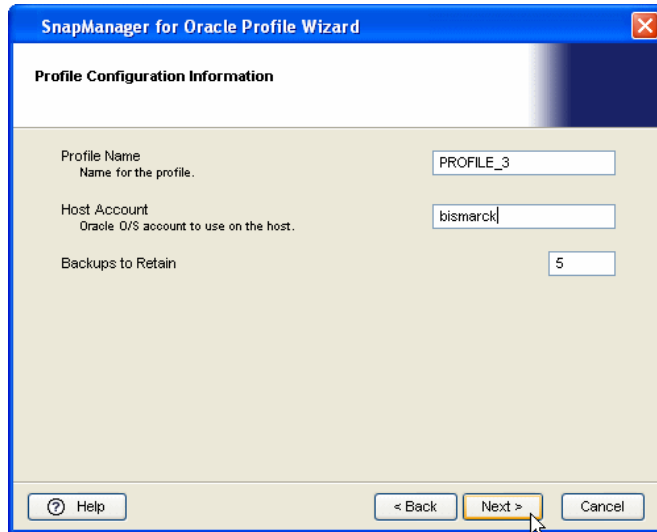
From the main GUI screen:

- ◆ Perform tasks in any one of three ways:
 - ❖ Select from the pull-down menus from the Menu bar at the top of the screen
 - ❖ Select the task from the task panel on the lower left
 - ❖ Select the repository, profile, backup or clone and then use the right-click menu
- ◆ Access the SnapManager for Oracle GUI help documentation from the toolbar.

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.



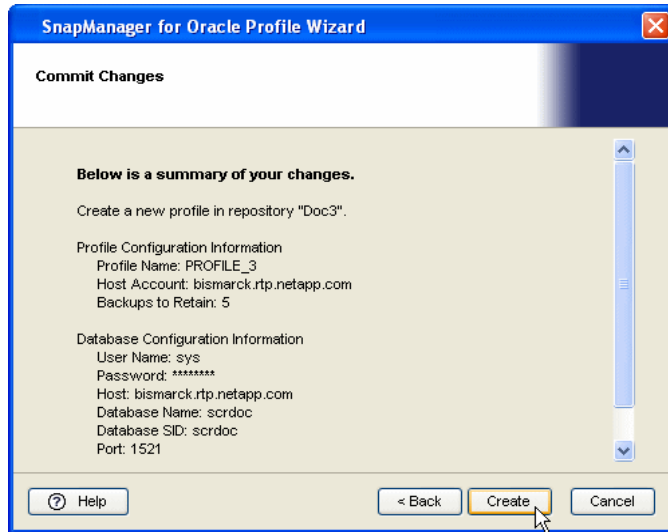
The screenshot shows a window titled "SnapManager for Oracle Profile Wizard" with a standard Windows-style title bar (blue with a red close button). The main content area is titled "Profile Configuration Information". It contains three configuration fields:

- Profile Name**: Labeled "Name for the profile.", with a text box containing "PROFILE_3".
- Host Account**: Labeled "Oracle O/S account to use on the host.", with a text box containing "bismarck".
- Backups to Retain**: A numeric spinner box set to "5".

At the bottom of the window, 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.

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 performed. If the information is correct, execute the task. Click the Back button to go back to a previous screen to enter correct information.



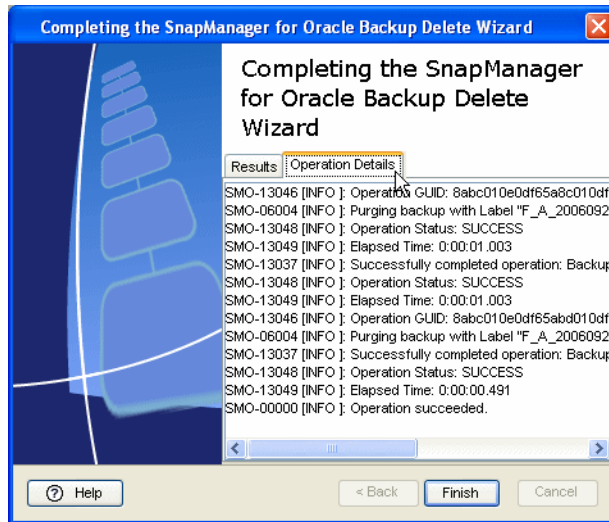
Viewing operation details

In any wizard, when you commit to the changes and execute the operation, a status page opens with the log messages. Adjust the level of detail that appears in the log messages by selecting, holding, and sliding the Output Level bar. There are four levels of information:

- ◆ Fatal errors
- ◆ Error messages
- ◆ Warning messages
- ◆ Complete information

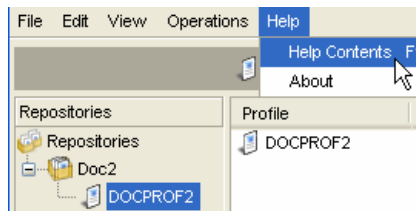
Placing the Output Level at the far right gives you the most complete information in the Log Messages window.

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. If it was unsuccessful, the log may contain information helpful in understanding what caused the failure.



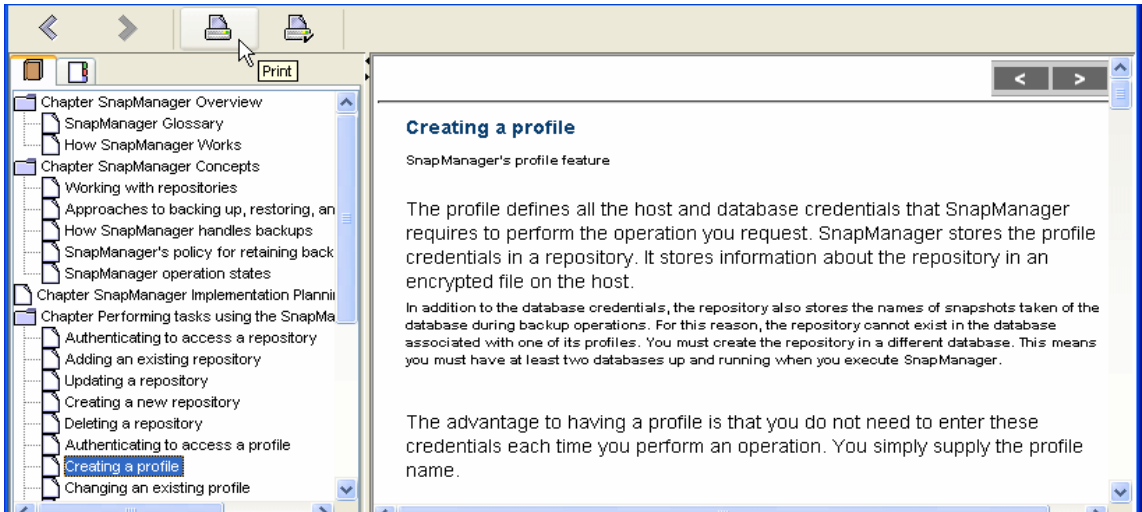
Accessing GUI help

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 chapter includes the following topics:

- ◆ [“Installing and upgrading SnapManager tasks”](#) on page 46
- ◆ [“Downloading the SnapManager software package”](#) on page 49
- ◆ [“Installing or upgrading the SnapManager software packages”](#) on page 51

Installing and upgrading SnapManager tasks

Before installing SnapManager, make sure that your system is set up correctly and has all the prerequisites installed. Refer to “[SnapManager Deployment Considerations](#)” on page 27 for more information about requirements and prerequisites.

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.

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 49. There is also information about the documentation available for these NetApp products in "Where to go for more information" on page 173.
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 49.
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 51.

Step	Action	Details
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 52.
Preparing to use SnapManager		
5	If installing SnapManager for the first time on this host environment, create a new repository for the profiles to use for the databases.	See the section “ Updating and creating repositories ” on page 60. If using the GUI, refer to the GUI online help.
6	Check to make sure all the required components are installed and working with SnapManager.	“ Verifying the environment ” on page 74.

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 49.
- ◆ 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. The installer automatically installs the CLI and the server software. If using the GUI, you must also download the GUI installer. You can use the GUI on Windows regardless of the host platform used.

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: <code>chmod 544 netapp.smo*</code>
4	If you are upgrading from SnapManager for Oracle 2.0, stop the server. <code>smo_server stop</code>
5	Use the following command to start the installation of the software on the host. For AIX, use: <code>./netapp.smo.aix-2.1.0.bin</code> For HP-UX, use: <code>./netapp.smo.hpux-2.1.0.bin</code> For Linux, use: <code>./netapp.smo.linux-2.1.0.bin</code> For Solaris, use: <code>./netapp.smo.sunos-2.1.0.bin</code> Result: The command installs the software in the: <code>/opt/NTAPsmo</code> directory for Solaris <code>/opt/NetApp/smo</code> directory for all other platforms
6	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.
7	At the prompt for a new server password, enter a password. This password is required when using SnapManager. Refer to “ Managing Credentials ” on page 112 for more information about credentials and passwords.

Step	Action
8	At the prompt for operating system user, press Enter to accept the default value. For a new installation, the default value for user is <code>oracle</code> .
9	At the prompt for operating system group, press Enter to accept the default value. For a new installation, the default value is <code>dba</code> .
10	At the Configuration Summary screen, press Enter to continue. If this is a new installation, the software is installed, including 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.
11	At the message Installation Complete, press Enter to exit the installer.

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 or another location on the system.
2	Download the <code>netapp.smo.windows-2.1.0.exe</code> file containing the SnapManager GUI to C:\Program Files or any location on the system.
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 .
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-2.1.0.

Step	Action
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 .

About this chapter

This chapter provides information about getting started using the CLI, the GUI, and creating and updating your repositories.

This chapter includes the following topics:

- ◆ [“Starting up the CLI or GUI”](#) on page 56
- ◆ [“Allowing access to SnapManager secured resources”](#) on page 57
- ◆ [“Updating and creating repositories”](#) on page 60
- ◆ [“Verifying, starting and stopping the server”](#) on page 64

Starting up the CLI or GUI

Starting the CLI

Use SnapManager for Oracle from the CLI by entering a command at the prompt on your UNIX host.

Starting the GUI

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

- ◆ Double-click the SnapManager for Oracle GUI 2.1.0 shortcut on your desktop.
- ◆ Select **Start > All Programs > NetApp > SnapManager for Oracle 2.1.0 > SnapManager for Oracle 2.1.0**.

Allowing access to SnapManager secured resources

About allowing users access to secured resources

Certain objects in SnapManager for Oracle are secured by passwords. As the administrator, you set the passwords for those resources and users set their credentials to access only those resources involved in the operations they execute.

Secured resources are:

- ◆ SnapManager hosts, local and remote
- ◆ Repositories
- ◆ Profiles

The credentials allow users to perform operations only on the hosts, repositories and profiles for which they have permissions.

Credentials are stored in the operating system user's home directory. Users do not have to re-enter the passwords when they access the resource again, unless the cache in the operating system user's home directory is cleared.

Administrators manage the passwords. Users manage their credentials.

What this means for GUI users

The SnapManager for Oracle GUI prompts for the server password, which users must enter before they can create profiles. Only those with administrator privileges can change host server passwords.

Another change from earlier versions is that when a profile is created or updated, the SnapManager GUI prompts for the profile password, instead of the database password. If you have upgraded from a version prior to SnapManager 2.1 for Oracle, the profile password defaults to the database password used in the previous version.

Users can change their profile password by modifying the profile properties page in the GUI.

What this means for CLI users

There are new `smo credential` commands and options to create and manage the credentials that allow users access to secured resources.

Creating the SnapManager host server password

As administrator, you are asked to create a server password for the local host during the SnapManager host software installation process.

Users who access that server have to enter the host server credentials the first time they execute an operation on any SnapManager for Oracle server. When cloning to a remote host, users must also enter the credentials for the target host.

To modify the password you created when you installed the SnapManager software, enter this command, as root:

```
smo_server config -password password
```

Users set their host credentials: Users can set their credentials to access a host using this command format:

```
smo credential set -host -name host_name -password password
```

They do not actually set the password for the host servers. They are setting the credentials for them to access the hosts for which you have given them privileges.

Creating the Repository password

You can change the repository password with `sysdba` privileges. Create the password using this command from the SQL prompt.

```
SQL> alter user username identified by new_repo_password
```

Users set their repository credentials: Users can set their credentials to access the repository using the command format:

```
smo credential set -repository -host host_name -dbname  
database_name -port port -login -username username -password  
password
```

Example:

```
smo credential set -repository -host bismarck -dbname repo1  
-username user1 -password simba
```

Output:

```
SMO-12345 [INFO ]: Updating credential cache for OS user "user3"  
SMO-12345 [INFO ]: Set repository credential for user "user3" on  
repo1@bismarck.  
Operation Id [Nff8080810da9018f010da901a0170001] succeeded.
```

Creating the profile password

Manage the profile password by updating an existing profile.

```
smo profile update -profile profile_name -profile-password password
```

Users set their profile credentials: Users set their credentials to access a profile using the command format:

```
smo credential set -profile -name profile_name -password password
```

Example:

```
smo credential set -profile -name profile -password 4j32x6
```

Output:

```
SMO-12345 [INFO ]: Updating credential cache for OS user "user1"  
SMO-12345 [INFO ]: Set profile credential for user "user1" on  
repol@bismarck.  
Operation Id [Nff8080810da9018f010da901a0170001] succeeded.
```

For information about clearing the credential cache and listing credentials for users, refer to “[Managing Credentials](#)” on page 112.

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 4K block size for the tablespace into which it is installed. Check the block size using the SQL command:

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

3. Enter credentials to update or create a repository.
4. Specify a name for the repository. This can be up to 30 characters.
5. 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.

About updating repositories

If you have no existing repositories, see “[Creating a repository procedure](#)” on page 63 to begin creating a repository for your database.

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.

Before updating existing repositories, run the `smo repository show` command to verify that there are no incomplete operations. If there are incomplete operations on the repository, abort the operations to free up all resources involved before updating.

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

Updating Repository considerations

In SnapManager 2.0 for Oracle and earlier, profile-to-repository mappings were stored on the SnapManager host server. To facilitate centralized management and multi-host operations, SnapManager 2.1 for Oracle stores the profile-to-repository mappings on the client side. This has a few implications for upgrades:

For the CLI user:

After the SMO server and repositories are upgraded, the user must run the `smo profile sync` command against the upgraded repository. This will pull in the profile-to-repository mappings for that repository into the user's cache. The user can then reference these profiles in any `smo` commands.

Since the profile-to-repository mapping cache is stored by OS user, the `smo profile sync` will need to be performed as each OS user that needs to reference the profiles.

Uniqueness of profile names:

Profile names in SnapManager 2.0 for Oracle and earlier must be unique within the scope of:

- ◆ Repository
- ◆ SnapManager host server

Profile names in SnapManager 2.1 for Oracle must be unique within the scope of:

- ◆ Repository
- ◆ OS user

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.

When you update existing repositories, the information for any existing clones is also updated based on the changes affecting clones in the SnapManager 2.1 for Oracle release. These changes are:

- ◆ The clone label attribute is set to the value of the `newsid` attribute. This is because a `-label` attribute is now available for clone naming. If left blank, the clone label is automatically generated by SnapManager
- ◆ The new `comment` attribute is empty.
- ◆ There is a field that describes if the backup, from which the clone was made, is part of the normal retention policy, or if it is designated to be retained forever. You can change this flag by updating the profile for the backup and entering the new retention policy you want. Refer to “[Changing an existing profile](#)” on page 119 for more information.

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 repo_dbname -host repo_host
    -login -username username
    -port repo_port
  [-force] [-noprompt]
  [-quiet | -verbose]:
```

Example: The following command line updates an existing repository.

```
smo repository update -repository -dbname SALESDB
-host server1 -login -username admin -port 1521
```

Updating the profile to repository mappings

To make the local user's home directory aware of the profile to repository mappings so it can route operation requests, use this command syntax.

```
smo profile sync -repository -dbname repo_dbname -host repo_host
-port repo_port -login -username repo_username
```

Example:

```
smo profile sync -repository -dbname smorepo -host bismarck -port
1521 -login -username user2
```

Output:

```
SMO-12345 [INFO ]: Loading profile mappings for repository
"user2@bismarck:smorepo" into cache for OS User "admin".
Operation Id [Nff8080810da9018f010da901a0170001] succeeded.
```

The repository and profiles are now accessible using the new version of SnapManager.

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 repo_dbname -host repo_host
    -login -username username
    -port repo_port
  [-force] [-noprompt]
  [-quiet | -verbose]
```

For details on the options and arguments used in this command, see [“The smo repository create command”](#) on page 238.

Example: The following command line creates a repository.

```
smo repository create -repository -dbname HRDB
-host server1 -login -username admin -port 1521
```

Verifying, starting and stopping the server

Accessing the server

To access the server, you must know the server password. You set the password during the installation process. For more information about managing these passwords, see [“Allowing access to SnapManager secured resources”](#) on page 57 and [“Managing Credentials”](#) on page 112.

Verifying the status of the server procedure

You must start the server before performing additional operations using SnapManager for Oracle. First, verify the status of the server.

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

```
smo_server status
```

Starting the server procedure

Before you can use the GUI or the CLI to initiate additional 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 60 to continue configuring SnapManager for Oracle.

Stopping the server

When you have finished using SnapManager, you may want to stop the server. To stop the server, use the following command as root:

```
smo_server stop
```

Restarting the server

To restart the server, use the following command as root.

```
smo_server restart
```

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 66
- ◆ [“Example procedure”](#) on page 68
 - [“Task 1: Verify the environment”](#) on page 68
 - [“Task 2: Create a profile”](#) on page 68
 - [“Task 3: Verify the profile configuration”](#) on page 69
 - [“Task 4: Back up your database”](#) on page 69
 - [“Task 5: Verify the backup”](#) on page 69
 - [“Task 6: Restore the backup”](#) on page 70
 - [“Task 7: Create a clone from the backup”](#) on page 70

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.

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
Determine which interface you will use	“ SnapManager Interfaces ” on page 35. Refer to the GUI online help for additional information about how to perform tasks using the GUI.
Download and install the SnapManager for Oracle software	“ Installing or Upgrading SnapManager ” on page 45
Configure credentials for hosts, repositories, profiles.	“ Allowing access to SnapManager secured resources ” on page 57
Update existing repositories to make the database schema and objects accessible, or create a new repository	“ About updating repositories ” on page 61 and “ Updating a repository procedure ” on page 62 “ Information required to update or create a repository ” on page 60 and “ Creating a repository procedure ” on page 63
Start the SnapManager server.	“ Verifying, starting and stopping the server ” on page 64.

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 -verbose
```

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 221, the “[The smo profile list command](#)” on page 228, and the “[The smo profile show command](#)” on page 229.

Examples:

The following command sets up a profile that uses ASM.

```
smo profile create -profile SALES1 -profile-password xyz  
-repository -dbname SNPMREPO  
-login -username admin1  
-port 1521 -host server1  
-database -dbname SALESDB  
-login -username admin2 -password oracle  
-port 1521 -host server2  
-osaccount oracle -osgroup osg  
-asm -host valiant -login -username asmuser -password asmpassword  
-port 1522 -sid asmsmo
```

Output:

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

This example adds the option to retain three backup copies.

```
smo profile create -profile SALES1 -profile-password ontap
-repository -dbname SNPMREPO
-login -username admin1
-port 1521 -host server1
-database -dbname SALESDB
-login -username admin2 -password oracle
-port 1521 -host server2
-osaccount oracle -osgroup group
-asm -host valiant -login -username asmuser -password asmpassword
-port 1522 -sid asmsmo
-retain 3
```

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 237.

```
smo profile verify -profile SALES1
```

Output:

Operation Id [Nd7169f567e8a4c6c9700b1ebac49993b] 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 180.

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

```
smo backup create -profile SALES1 -full -offline
-label full_backup_sales_May -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 200.

Example: The following command line verifies the backup.

```
smo backup verify -profile SALES1 -label full_backup_sales_May
-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 command](#)” on page 190.

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

```
smo backup restore -profile SALES1 -label full_backup_sales_May -
complete -recover -alllogs
```

Output:

```
Operation Id [8abc01ec0e78f3e2010e78f3fdd00001] succeeded.
```

Task 7: Create a clone from the backup

Use the `smo clone create` command to clone a backup of a database. For details on the options and arguments used in this command, see “[The smo clone create command](#)” on page 202.

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 early releases of SnapManager for Oracle and are not backwards compatible. SnapManager 2.0 and higher use clone specifications. 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 95.

2. The following example creates a clone from a backup using an existing clone specification, called `Sales1_clonespec.xml`, which is located on the host.

```
smo clone create -profile SALES1 -backup-label
full_backup_sales_May -newsid CLONE -label sales1_clone -clonespec
/opt/NetApp/smo/clonespecs/sales1_clonespec.xml
```

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 74
- ◆ [“Creating a profile”](#) on page 75
- ◆ [“Verifying the profile”](#) on page 79

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 251.

Example:

```
smo system verify -verbose
```

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.
```

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 76.

When creating a profile, you have a choice of methods to connect and interface with the database. Choose either:

- ◆ User name and password
- ◆ OS authentication

For the OS authentication, you enter the OS account user and group information. You can use OS authentication with Oracle authentication. For example, use OS authentication for the database and Oracle authentication for the ASM instance, or any combination of the two. However, you *cannot* use OS authentication for the database if the database is RAC.

The basic steps for creating or updating 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 username and password.
 - ❖ 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 and password required to connect to the database or the OS authentication information.
 - ❖ 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 and password required to connect to ASM or use the OS authentication information.
 - ❖ 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 RMAN, specify that you will use either:
 - ❖ Control files or the RMAN catalog.
 - ❖ To use the RMAN catalog, you must provide:
 - * The user name and password required to connect to the RMAN catalog or use the OS authentication information.
 - * 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. Backups that have been created with the `-retain-forever` option will not be deleted by the retention policy. See [“SnapManager’s policy for retaining backups”](#) on page 21 for information about how SnapManager retains backups.

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 profile_name -profile-password profile_password
-repository
  -dbname repo_dbname -host repo_host
  -login -username repo_username
  -port repo_port
-database
  -dbname db_dbname -host db_host
  -login -username db_username -password db_password
  [-sid db_sid] -port db_port
-osaccount account_name -osgroup osgroup
```

```

[-rman
  {-tnsname tnsname
  -login -username rman_username -password rman_password} |
  -controlfile]
[-asm
  -dbname asm_dbname
  -host asm_host
  -login -username asm_username -password asm_password
  -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 221, the “[The smo profile list command](#)” on page 228, and the “[The smo profile show command](#)” on page 229.

Examples of creating profiles

Examples:

The following command line sets up a profile that uses ASM and RMAN, using OS authentication for both the database and the ASM instance, and retaining 5 backups.

```

smo profile create -profile SALES1 -profile-password xyz123
-repository -dbname SNPMREPO -host server1 -login -username admin1
-port 1521 -database -dbname SALESDB
-host server2
-asm -dbname ASMDB -host server3 -sid ASMDB1
-rman -controlfile -osaccount oracleuser1 -osgroup dba
-retain 5

```

The following command line creates a profile that uses ASM and RMAN using username and password authentication.

```

smo profile create -profile SALES2 -profile-password xyz456
-repository -dbname SNPMREPO2
-login -username admin1
-port 1521 -host server1
-database -dbname SALESDB2
-login -username admin2 -password oracle
-osaccount oracle -osgroup osg
-port 1521 -host server2

```

```
-asm -host server3 -login -username asmuser -password asmpassword  
-port 1522 -sid asmsmo  
-rman -controlfile  
-retain 5  
-verbose
```

Output:

Operation Id [8abc01ec0e78ebda010e78ebe6a40005] succeeded.

This example adds the option to retain three backups. (The retain option applies only to successful backups that have not been used to create a clone and designated to be part of the normal retention policy. Regardless of the value for retain, SnapManager does not automatically remove backups that were used to create a clone or are designated to be retained forever. Refer to [“SnapManager’s policy for retaining backups”](#) on page 21 for more information.)

This is the same example without the ASM and RMAN information, using OS authentication.

```
smo profile create -profile SALES3 -profile-password xyz456  
-database -dbname SALESDB3  
-login -username admin3 -password oracle  
-osaccount oracle -osgroup dba3  
-port 1521 -host server3  
-retain 3
```

Output:

Operation Id [8abc01ec0e79ebda010e79ebe6a40005] succeeded.

**Example of
Updating a profile**

To update or modify a profile, use the `smo profile update` command. The following example changes the login information for the database described by the SALES1 profile. You would run this command *after* changing the login information on the database itself.

```
smo profile update -profile SALES1 -database -dbname SALESDB  
-sid SALESDB -login -username admin2 -password d4jPe7bw -port 1521  
-host server1
```

Output:

Operation Id [8abc01ec0e78ec33010e78ec3b410001] 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 SALES1
```

Output:

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

More information about managing profiles

For information about changing existing profiles or deleting profiles, see the [“Changing an existing profile”](#) on page 119 and [“Deleting profiles”](#) on page 120.

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 82
- ◆ [“Verifying a backup”](#) on page 85
- ◆ [“Restoring and recovering a database backup”](#) on page 87

Creating database backups

SnapManager gives you the capability to create backups of entire databases and portions of databases. The backups are virtual copies of the database 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 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 12.

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.
 - ❖ (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.

To make identifying the backup easier, 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.

(Optional) Enter the `-force` option if you want SnapManager to force the database into the appropriate state.

To take an online backup of a RAC database, the primary database instance can be MOUNTED if any remote instance is OPEN. Use `-force` if the local instance is SHUTDOWN, or no instance is OPEN. If the local instance is SHUTDOWN and at least one instance is OPEN, then using `-force` changes the local instance to MOUNTED. If no instance is OPEN, using `-force` changes the local instance to OPEN.

4. (Optional) To exclude a backup from being automatically deleted by the retention policy, enter the option `-retain-forever` when creating the backup. Specifying no retention option is the same as using the `-retain-normal` option. You may update the backup at a later time to include it in the normal retention policy using the `smo backup update` command with the `-retain-normal` option. For more information on how SnapManager retains backups, see [“SnapManager’s policy for retaining backups”](#) on page 21 and [“The smo backup create command”](#) on page 180.
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 179.

Creating a backup procedure

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

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

For details on the options and arguments used in this command, see [“The smo backup create command”](#) on page 180.

Examples:

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

```
# smo backup create -offline -tablespaces PSAPSR3USR -profile
SALES1 -label offline_tablespaces -force
```

Output:

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

This is an example of creating a full online backup that is exempt from being deleted by the retention policy.

```
# smo backup create -online -full -profile FIN1
-label online_full -retain-forever -force
```

Output:

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

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

```
# smo backup create -online -files
/oracle/QA1_sapdata/sapdata1/sr3_1/sr3.data1 -profile HR1 -label
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. Perform the verification immediately after making the backup, or 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 177.)
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 179.

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

For details on the options and arguments used in this command, see [“Using the smo backup verify command”](#) on page 200.

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

```
# smo backup verify -profile SALES1 -label full_backup -quiet
```

Output:

```
DBVERIFY - Verification starting : FILE =
/opt/NetApp/smo/mnt/_mnt_triggerfish_bismarck_nfs_smoke_SMO_SALES1
_NFSTEST_BISMAR_NFSTEST_F_C_0_8abc01ec0e78f05c010e78f06f120001/dat
afiles/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_bismarck_nfs_smoke_SMO_/SALES
1_NFSTEST_BISMAR_NFSTEST_F_C_0_8abc01ec0e78f05c010e78f06f120001/da
tafiles/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 12.

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 177.)
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 command”](#) on page 190.

Example: This example restores the entire backup labeled `ProdFriday` that is part of the profile `Production`. It also recovers all logs.

```
smo backup restore -profile Production -label ProdFriday  
-complete -recover -alllogs
```


About this chapter

This chapter details how to use SnapManager to clone backups of databases.

This chapter includes the following topics:

- ◆ “[General SnapManager database cloning information](#)” on page 92
- ◆ “[Creating a clone specification](#)” on page 95
- ◆ “[Cloning a database from a backup](#)” on page 103
- ◆ “[Cloning a database in its current state](#)” on page 105
- ◆ “[Cloning a database to an alternate host](#)” on page 106
- ◆ “[Options for cloning RAC databases](#)” on page 108

General SnapManager database cloning information

SnapManager's clone feature

SnapManager's clone feature lets you quickly create a copy of a database 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 space only for the changes between the backup and the clone. The clone is a writable snapshot of the original database. 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 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

For more information about the command, see “[The smo clone create command](#)” on page 202. You must have a clone specification defined, or use the `smo clone template` command to create a template clone specification, and edit it. Refer to “[Creating a clone specification](#)” on page 95 to see the information the clone specification contains.

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. To create a physical copy, you must have space available that is equal to the current size of the database. For example, if your database 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 while it is offline or online.
2. Create a clone specification file for the database. SnapManager uses this file to create the clone based on the information in the original database 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. You can clone to the same host on which the database resides, or to an alternate host.

Note

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. You can have a clone on a different host using the same SID. As a result, there is a new `-label` option to designate a unique name for the clone. If you do not use this option, SnapManager creates a unique name for the clone that includes the SID, date and time.

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:

```
initclone_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 as well as a non-clustered database. A RAC clone starts as a single database. For more information, see “[Options for cloning RAC databases](#)” on page 108.

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 `initclone_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:

`spfileclone_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 new clone.

Note

SnapManager 2.0 for Oracle and higher uses clone specifications, which are not backwards compatible with clone maps from prior releases. You must change the clone map .txt file from a release prior to SnapManager 2.0 for Oracle to an XML 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 disk groups, raw devices) that contain data files in the source database. There is no need to map storage components that just contain control files and archived logs.

◆ 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 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 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 must change to create the clone, the clone operation is aborted and you receive an error message.

Note

Directories specified for certain Oracle parameters in the clone will be destroyed when the clone is deleted, and should only contain data for the cloned database: Archive Log Destination, Background, Core, and User Dump Destinations.

```

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

```

◆ Oracle home and operating system account information

Enter this information in the clone specification:

- ❖ If you are cloning from a source database host to a target host, and you want the new clone to use the information from the target host. The default behavior is for the clone on the target host to use the source host's Oracle home, OS user and group name specifications.
- ❖ If you want to change these values from the default (the current values on the local host).

```

<oracle_home>
  /mnt/oracle_home
</oracle_home>

<oracle_os_account>
  <username>oracle</username>
  <group>dba</group>
</oracle_os_account>

```

Example of a completed database specification

The following is an example of a database specification structure, including control files, redo logs, Oracle parameters, home and OS account information.

```

<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>
  <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>

```

Table continued on next page...

(Table continued)

```
<oracle_home>
    /mnt/oracle_home
</oracle_home>
<oracle_os_account>
    <username>oracle</username>
    <group>dba</group>
</oracle_os_account>
</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_01.dbf'
    SIZE 41943040 REUSE AUTOEXTEND ON NEXT 655360
    MAXSIZE 32767M
  </sql-statement>
</sql-statements>
```

Clone a backup of a database using the clone specification. Refer to “[Cloning a database from a backup](#)” on page 103, “[Cloning a database in its current state](#)” on page 105, or “[Cloning a database to an alternate host](#)” on page 106 for information about creating clones.

Cloning a database from a backup

General steps for cloning a database

There are two options when cloning a database. You can:

- ◆ Clone the database from its current state, which performs the backup and cloning with one command on the same host or to a different host. Refer to [“Cloning a database in its current state”](#) on page 105 and [“Cloning a database to an alternate host”](#) on page 106 for more information.
- ◆ Clone from an existing backup on the same host or a different host.

This section takes you through the basic steps to clone a database from a backup. Follow this procedure whether you are cloning to the same host or to an alternate host. For a syntax example, see [“Cloning a database from a backup procedure”](#) on page 104.

The basic steps for cloning a database 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 177.)
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 for the new RAC database. See [“Cloning with RAC databases”](#) on page 108 for more information.
 - ❖ 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. However, you can have a database on another host with the same SID.

3. Specify a label for the new clone. If this option is not specified, SnapManager generates a label for the clone. The generated label includes the SID, date and time.
4. Specify the target host if cloning the database to an alternate host.
5. Specify the name of the profile from which the backup was created.

6. 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 a template with the CLI which you can edit, before you start the clone operation.

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

Note

If using TNSNAMES, add the database to the TNSNAMES file on the client machines that need to access the new database.

Cloning a database from a backup procedure

To use the CLI `smo clone create` command to clone a backup of a database, 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 95.
2	Enter the <code>smo clone create</code> command using this syntax. <pre>smo clone create -backup-id <i>backup_guid</i> -backup-label <i>backup_label_name</i> -newsid <i>new_sid</i> -label <i>clone_label</i> -profile <i>profile_name</i> -clonespec <i>full_path_to_clonespecfile</i></pre> For details on the options and arguments used in this command, see “The smo clone create command” on page 202. Example: The following example creates a clone from a backup labeled <code>Test1</code>, which is included in the profile named <code>alltests</code>. The clone uses the clone specification file called <code>clonespec.xml</code>. <pre>smo clone create -backup-label Test1 -newsid EXC79D -label clone1 -profile alltests -clonespec ./maps/clonespec.xml</pre>

Cloning a database in its current state

Cloning a database in its current state

You can create a backup and clone in one procedure from the current state of the database. Instead of specifying a backup to be used for the clone using either the `-backup-id` or `-backup-label` flags, you specify the profile with the flag `-current`, which tells SnapManager 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 -label 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 a database to an alternate host

Prerequisites

There are prerequisites that must be set up before you can clone to a host other than the host on which the database resides. The following table shows the requirements for the source and target host set up. Verify all supported requirements and versions with the SnapManager & SnapDrive Compatibility Matrix on the NOW site at <http://now.netapp.com/NOW/knowledge/docs/docs.shtml> in the Interoperability section.

Prerequisite set up	Requirement
Architecture	Must be the same on both the source and target hosts
Operating system and version	Must be the same on both the source and target hosts
SnapManager 2.1 for Oracle	Must be installed and running on both the source and target hosts
Credentials	Must be set for the user to access the target host
Oracle	The same software version must be installed on both the source and target hosts. The Oracle Listener must be running on target host.
Compatible storage stack:	Must be the same on both the source and target hosts
❖ Protocol used to access data files	Must be the same on both the source and target hosts
❖ Volume managers	Must be configured on both the source and target hosts and must be compatible versions

Cloning a database to an alternate host procedure

This section takes you through the basic steps to clone a database to an alternate host from a backup.

This procedure assumes that you already have a profile and have made a full, backup to clone to another host.

To use the CLI `smo clone create` command to clone a backup of a database on a source host to an alternate target host, 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 95.
2	Enter the <code>smo clone create</code> command using this syntax. <pre>smo clone create -backup-label <i>backup_label_name</i> -newsid <i>new_sid</i> -host <i>target_host</i> -label <i>clone_label</i> -comment -profile <i>profile_name</i> -clonespec <i>full_path_to_clonespecfile</i></pre> For details on the options and arguments used in this command, see “The smo clone create command” on page 202. Example: The following example creates a clone from a backup labeled <code>Test1</code>, which is included in the profile <code>alltests</code> on a source host. The clone is labeled as <code>clone1</code> and uses the clone specification file called <code>clonespec.xml</code>. The clone resides on the target host, named <code>host1</code>. <pre>smo clone create -backup-label Test1 -newsid EXC79D -host host1 -label clone1 -profile alltests -clonespec ./maps/clonespec.xml</pre>

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 108.

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 108.

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 -backup-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 database, complete the following steps.

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

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

Note

You must undo all the steps you performed to turn this clone into a RAC database before attempting to delete the clone.

Example of setting up a cloned database as a one node RAC instance

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 -backup-label rac_full_backup -profile
rac_profile -newsid CLONE1 -clonespec rac_nfs_clonespec.xml
-verbose
$ export ORACLE_SID=CLONE1
$ sqlplus
      shutdown immediate;
$ vi initCLONE11.ora
      *.cluster_database = TRUE
      CLONE11.instance_number=1
      CLONE11.thread=1
      CLONE11.undo_tablespace='UNDOTBS1'
$ srvctl add database -d CLONE1 -o
/u02/app/oracle/product/10.2.0/db -m rtp.company.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:

- ◆ “[Managing Credentials](#)” on page 112
- ◆ “[Aborting an operation](#)” on page 114
- ◆ “[Issuing SnapManager CLI Commands from an alternate host](#)” on page 117
- ◆ “[Deleting repositories](#)” on page 118
- ◆ “[Changing an existing profile](#)” on page 119
- ◆ “[Deleting profiles](#)” on page 120
- ◆ “[Listing backups](#)” on page 121
- ◆ “[Showing backup details](#)” on page 123
- ◆ “[Updating the backup retention policy](#)” on page 127
- ◆ “[Deleting backups](#)” on page 129
- ◆ “[Mounting and unmounting backups](#)” on page 131
- ◆ “[Listing clones](#)” on page 134
- ◆ “[Showing clone details](#)” on page 135
- ◆ “[Deleting clones](#)” on page 137
- ◆ “[Checking the version of SnapManager](#)” on page 138
- ◆ “[Uninstalling SnapManager software](#)” on page 139

Managing Credentials

About the Credential command

The following is the general format for the commands to clear, or list credentials.

```
smo credential clear | list
    -quiet | -verbose
```

For more information about setting credentials, refer to [“Allowing access to SnapManager secured resources”](#) on page 57. For more information about the `smo credential` commands, refer to [“SnapManager Command Reference”](#) on page 175.

Clearing credentials

To clear the credentials for a user use this command format.

```
smo credential clear
```

Example:

```
smo credential clear
```

Output:

```
SMO-20024 [INFO ]: Cleared credentials for user "user1".
```

Listing credentials

To list credentials for a user, use this command.

```
smo credential list
    [-quiet | -verbose]
```

Example:

```
smo credential list
```

Output:

```
Credential cache for OS user "user1":
```

```
Repositories:
```

```
    hancock_smoke_user@SMOREPO/hotspur:1521
    bismarck_smoke_user@SMOREPO/hotspur:1521
    user1_1@SMOREPO/hotspur:1521
```

```
Profiles:
```

```
    HSDBR (Repository: philipc_2_1@SMOREPO/hotspur:1521)
    PBCASM (Repository: philipc_2_1@SMOREPO/hotspur:1521)
```

```
HSDB (Repository: hancock_smoke_user@SMOREPO/hotspur:1521)
[PASSWORD NOT SET]
Hosts:
  bismarck
  saratoga
  revenge
  hancock
```

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 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 23.

- ◆ **(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 database and/or on the storage systems.
- ❖ To unlock the database for other SnapManager operations. For example, suppose that the backup operation put a tablespaces 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 items 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 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 115.

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 177.)
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 179.

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 operation_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 217.

Issuing SnapManager CLI Commands from an alternate host

You can issue CLI commands from a host other than the database host. For the system to dispatch an operation to the correct host, it must first know where to find the profile for the operation. In this procedure the system keeps the profile to repository mapping information for a file in the user's home directory on the local host. To make the local user's home directory aware of the profile to repository mappings so it can route the operation request, use this command syntax.

```
smo profile sync -repository -dbname repo_dbname -host repo_host
-port repo_port -login -username repo_username
```

Example:

```
smo profile sync -repository -dbname smorepo -host bismarck -port
1521 -login -username user2
```

Output:

```
SMO-12345 [INFO ]: Loading profile mappings for repository
"user2@bismarck:smorepo" into cache for OS User "admin".
Operation Id [Nff8080810da9018f010da901a0170001] succeeded.
```

Now you can enter commands against profiles in the synchronized repository and they are routed to the appropriate host.

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 -port
   port_number -host host}
  [-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 -host host1 -port 1521
```

For more information about this command, see [“The smo repository delete command”](#) on page 240.

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
[-profile-password profile_password]
-database
  -dbname db_dbname -host db_host
  -login -username db_username -password db_password
  [-sid db_sid] -port db_port
  -osaccount account_name
  -osgroup group
[-rman
  {-tnsname tnsname
  -login -username rman_username -password rman_password} |
  -controlfile]
[-asm
  -dbname asm_dbname
  -host asm_host
  -login -username asm_username -password asm_password
  -port asm_port -sid asm_sid]
[-retain number]
[-remove-asm]
[-remove-rman]
[-quiet | -verbose]
```

For details on the options and arguments used in this command, see [“The smo profile update command”](#) on page 233.

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 -profile-password 4j362x
-repository -dbname database1 -login -username sys -password
pacific -host ocean -port 1521
```

Deleting profiles

You can delete a profile anytime, as long as it does not contain successful backups. To delete a 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 226

Example: The following command line deletes the profile MYTEST.

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

Listing backups

Listing current backups in profiles

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:

Backup list displays all backups for a specified profile. It lists the following information:

- ◆ Start Date and time of the backup
- ◆ Label for the backup
- ◆ Status of the backup
- ◆ Scope of the backup
- ◆ Retention Policy for the backup
- ◆ ID for the backup

If the `-verbose` option is specified, it also lists the following information:

- ◆ Comment entered when creating the backup

General steps for listing backup information

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

1. Specify the name of the profile associated with the backups 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]
```

Example:

```
smo backup list -profile SALES1 -verbose
```

Output:

Start Date	Label	Status	Scope	Mode	Retention	ID	Comment
2007-03-20 21:12:15	bkup13	SUCCESS	PARTIAL	OFFLINE	NORMAL	ff8...30001	test comment
2007-03-20 21:10:11	fail2	FAILED	PARTIAL	ONLINE	N/A	ff8...20001	
2007-03-20 21:06:22	fail1	FAILED	PARTIAL	OFFLINE	N/A	ff8...70001	
2007-03-20 21:04:27	bkup12	SUCCESS	FULL	ONLINE	NORMAL	ff8...270001	
2007-03-16 18:54:02	bkup11	SUCCESS	FULL	ONLINE	NORMAL	ff8...d20001	
2007-03-15 21:47:11	bkup8	SUCCESS	FULL	ONLINE	NORMAL	ff8...e0001	
2007-03-15 21:38:43	label5	SUCCESS	FULL	ONLINE	NORMAL	ff8...50001	
2007-03-15 21:20:40	label4	FAILED	FULL	OFFLINE	N/A	ff8...a0001	
2007-03-15 20:44:55	bkup3	FAILED	FULL	OFFLINE	N/A	ff8...820001	
2007-03-15 20:41:19	label2	FAILED	FULL	OFFLINE	N/A	ff8...80001	
2007-03-15 20:39:55	label1	SUCCESS	FULL	ONLINE	NORMAL	ff8...10001	

For details on the options and arguments used in this command, see “[The smo backup list command](#)” on page 186.

Showing backup details

Displaying detailed information about a backup

You can display detailed information about a particular backup in a profile by using the `smo backup show` command.

This operation displays the following information for each backup:

- ◆ The backup ID
- ◆ Whether the backup succeeded or failed
- ◆ Backup type
- ◆ Backup mode
- ◆ Mount status
- ◆ The backup label
- ◆ Comment
- ◆ The date and time the operation started and ended
- ◆ Information about whether the backup was verified
- ◆ The backup retention policy
- ◆ The database and host name
- ◆ The SCN
- ◆ The tablespaces and data files from the database backed up
- ◆ What type of backup was performed - full, partial, online, offline
- ◆ The storage system and volumes where the files are located
- ◆ The snapshots taken and their location

If the `-verbose` option is specified, this additional information is shown:

- ◆ The clones made from the backup, if there are any
- ◆ Verification information

General steps for showing backup details

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.
2. Specify the label or id of the backup

Showing backup details procedure

To display detailed information about a particular backup, use the `smo backup show` command. It has the following general format:

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

Example:

```
smo backup show -profile SALES1 -label BTNFS
```

Output:

```
Backup id: 8abc013111a450480111a45066210001
Backup status: SUCCESS
Backup type: FULL
Backup mode: OFFLINE
Mount status: NOT MOUNTED
Backup label: BTNFS
Backup comment:
RMAN Tag: SMO_BTNFS_1175283108815
Backup start time: 2007-03-30 15:26:30
Backup end time: 2007-03-30 15:34:13

Verification status: OK
Backup Retention Policy: NORMAL
Backup database: hsd1
    SCN: 2700620
    Tablespace: SYSAUX
        Datafile: /mnt/weever/data/hsdb/sysaux01.dbf [ONLINE]
    Tablespace: USERS
        Datafile: /mnt/weever/data/hsdb/users01.dbf [ONLINE]
    Tablespace: UNDOTBS1
        Datafile: /mnt/weever/data/hsdb/undotbs01.dbf [ONLINE]
Host: hancock
    Filesystem: /mnt/weever/data
        File: /mnt/weever/data/hsdb/SMOBakCtl_1175283005231_0
        File: /mnt/weever/data/hsdb/SMOBakCtl_1175283007914_2
        File: /mnt/weever/data/hsdb/undotbs02.dbf
    Filesystem: /mnt/weever/logs
        File:
/mnt/weever/logs/HSDB/archivelog/2007_03_29/o1_mf_1_36_30pw1kvd_.a
rc
        File:
/mnt/weever/logs/HSDB/archivelog/2007_03_30/o1_mf_1_46_30t1s445_.a
rc
        File:
/mnt/weever/logs/HSDB/archivelog/2007_03_30/o1_mf_2_62_30tmk567_.a
rc
```

```

Storage System: weever.rtp.netapp.com
  Volume: hs_logs
    Snapshot:
      SMO_HSDBR_hbdb1_F_C_1_8abc013111a450480111a45066210001_0
        File:
          /mnt/weever/logs/HSDB/archivelog/2007_03_29/o1_mf_1_36_30pw1kvd_.a
          rc
            File:
              /mnt/weever/logs/HSDB/archivelog/2007_03_30/o1_mf_1_46_30tls445_.a
              rc
                File:
                  /mnt/weever/logs/HSDB/archivelog/2007_03_30/o1_mf_2_62_30tmk567_.a
                  rc
      Volume: hs_data
        Snapshot:
          SMO_HSDBR_hbdb1_F_C_1_8abc013111a450480111a45066210001_0
            File: /mnt/weever/data/hsdb/SMOBakCtl_1175283005231_0
            File: /mnt/weever/data/hsdb/SMOBakCtl_1175283007914_2
            File: /mnt/weever/data/hsdb/undotbs02.dbf

```

If you add the `-verbose` option to this command, in addition to the previous output, you will get the following information.

```

CLONES:
  Clone id: 8abc013111b916e30111b916ffb40001
  Clone status: SUCCESS
  Clone SID: hsdbc
  Clone label: hsdbc
  Clone start time: 2007-04-03 16:15:50
  Clone end time: 2007-04-03 16:18:17
  Host: hancock
  Filesystem: /mnt/weever/data_clone
    File: /mnt/weever/data_clone/hsdb/users01.dbf
    File: /mnt/weever/data_clone/hsdb/undotbs02.dbf
    File: /mnt/weever/data_clone/hsdb/undotbs01.dbf
    File: /mnt/weever/data_clone/hsdb/sysaux01.dbf
VERIFIES:
  Verify id: 8abc013111b919c10111b919da700001
  Backup status: SUCCESS
  Backup start time: 2007-04-03 16:18:58
  Backup end time: 2007-04-03 16:21:12
  Verify result: OK
  Verify output: DBVERIFY - Verification starting : FILE =
  /u/user1/p4/capstan/2.1/mnt/_mnt_weever_data_SMO_HSDBR_hbdb1_F_C_1
  _8abc013111a450480111a45066210001/hsdb/sysaux01.dbf

```

```
...
DBVERIFY - Verification complete

Total Pages Examined          : 49920
Total Pages Processed (Data)  : 11669
Total Pages Failing   (Data)  : 0
Total Pages Processed (Index): 10279
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 16180
Total Pages Processed (Seg)   : 0
Total Pages Failing   (Seg)   : 0
Total Pages Empty           : 11792
Total Pages Marked Corrupt    : 0
Total Pages Influx          : 0
Highest block SCN            : 2700634 (0.2700634)
...
```

For details on the options and arguments used in this command, see “[The smo backup show command](#)” on page 193.

Updating the backup retention policy

About updating backups

There are two reasons to update a backup.

- ◆ To change the backup so it is considered eligible for deletion by the backup retention policy
- ◆ To change the backup so it is not considered eligible for deletion by the backup retention policy

Changing the backup so it is considered for deletion by the retention policy

To make a backup that was previously specified to be retained forever eligible for deletion by the retention policy, use this command format:

```
smo backup update  
{-id backup_guid | -label backup_label}  
-profile profile_name  
-retain-normal
```

Changing the backup so it is not considered for deletion by the retention policy

To update a backup so it is no longer considered for deletion after it has been created as eligible for the normal retention policy, use this command format:

```
smo backup update  
{-id backup_guid | -label backup_label}  
-profile profile_name  
-retain-forever
```

Deleting the backup you created as exempt from the retention policy

To delete a backup you created as exempt from the retention policy, you must first update the backup to make it eligible for deletion. Follow this procedure to update and then delete the backup.

Step	Action
1	To update the backup, use this command format. <pre>smo backup update {-id <i>backup_guid</i> -label <i>backup_label</i>} -profile <i>profile_name</i> -retain-normal</pre>
2	To delete the backup, use this command format. <pre>smo backup delete {-label <i>label_name</i> -id <i>backup_guid</i> } -profile <i>profile_name</i></pre>

Deleting backups

Using the delete backups feature

You can delete backups for a database, if the backup has not been used to create a clone and the backup is set to be part of the normal retention policy. 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 21.

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 130.

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 the name of the profile from which the backup was created.
3. (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 184.

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 132.

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 177.)
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 179.

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_guid |-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 188.

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

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

If you use the `-force` option, SnapManager 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 198.

Examples:

The following is an example of a mount operation.

```
# smo backup mount -label test -profile SALES1 -verbose
```

Output:

```
SMO-13046 [INFO ]: Operation GUID 8abc013111b9088e0111b908a7560001
starting on Profile SALES1

SMO-08052 [INFO ]: Beginning to connect mount(s) [/mnt/weever/logs,
/mnt/weever/data] from logical snapshot
SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001.

SMO-08025 [INFO ]: Beginning to connect mount /mnt/weever/logs from
snapshot SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001_0
of volume hs_logs.

SMO-08027 [INFO ]: Finished connecting mount /mnt/weever/logs from
snapshot SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001_0
of volume hs_logs.
```

```
SMO-08025 [INFO ]: Beginning to connect mount /mnt/weever/data from
snapshot SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001_0
of volume hs_data.

SMO-08027 [INFO ]: Finished connecting mount /mnt/weever/data from
snapshot SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001_0
of volume hs_data.

SMO-08053 [INFO ]: Finished connecting mount(s) [/mnt/weever/logs,
/mnt/weever/data] from logical snapshot
SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001.

SMO-13037 [INFO ]: Successfully completed operation: Backup Mount
SMO-13048 [INFO ]: Operation Status: SUCCESS
SMO-13049 [INFO ]: Elapsed Time: 0:01:00.981
Operation Id [8abc013111b9088e0111b908a7560001] succeeded.
```

The following is an example of an unmount operation:

```
# smo backup unmount -label test -profile SALES1 -verbose
```

Output:

```
SMO-13046 [INFO ]: Operation GUID 8abc013111b909eb0111b90a02f50001
starting on Profile SALES1
SMO-08028 [INFO ]: Beginning to disconnect connected mount(s)
[/u/user1/p4/capstan/2.1/mnt/_mnt_weever_logs_SMO_SALES1_hbdb1_F_C
_1_8abc013111a450480111a45066210001,
/u/user1/p4/capstan/2.1/mnt/_mnt_weever_data_SMO_SALES1_hbdb1_F_C
_1_8abc013111a450480111a45066210001].
SMO-08030 [INFO ]: Done disconnecting connected mount(s)
[/u/user1/p4/capstan/2.1/mnt/_mnt_weever_logs_SMO_SALES1_hbdb1_F_C
_1_8abc013111a450480111a45066210001,
/u/user1/p4/capstan/2.1/mnt/_mnt_weever_data_SMO_SALES1_hbdb1_F_C
_1_8abc013111a450480111a45066210001].
SMO-13037 [INFO ]: Successfully completed operation: Backup Unmount
SMO-13048 [INFO ]: Operation Status: SUCCESS
SMO-13049 [INFO ]: Elapsed Time: 0:00:33.715
Operation Id [8abc013111b909eb0111b90a02f50001] succeeded.
```

Listing clones

Listing clones procedure

The information you can view about the clones in a profile includes:

- ◆ The ID for the clone
- ◆ Status of the clone operation
- ◆ Oracle SID for the clone
- ◆ Host on which the clone resides
- ◆ Label for the Clone

If the `-verbose` option is specified, the output also shows:

- ◆ Any comment entered for the clone

To display a list of all clones for a profile, use this command format.

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

Examples:

```
smo clone list -profile NFSDB_REVENGE -verbose
```

Output:

ID	Status	SID	Host	Label	Comment
-----	-----	-----	-----	-----	-----
8ab...01	SUCCESS	hsdbc	server1	back1clone	

This example shows the output for a profile containing no clones to list.

```
smo clone -list -profile NFSDB_VALIANT -verbose
```

Output:

There are no clones for this profile.

For more information about listing clones, see “[The smo clone list command](#)” on page 205.

Showing clone details

About displaying clone details

You can display the following information about a specific clone.

- ◆ Clone SID
- ◆ Clone GUID
- ◆ Clone operation status
- ◆ Clone create start date/time
- ◆ Clone create end date/time
- ◆ Clone label
- ◆ Clone comment
- ◆ Backup label
- ◆ Backup ID
- ◆ Source database
- ◆ Backup start time
- ◆ Backup end time
- ◆ Database name
- ◆ Database tablespaces and data files
- ◆ Host name
- ◆ Host file systems containing data files
- ◆ Storage Systems
- ◆ Storage system volumes and snapshots backing the clone

Showing clone details procedure

To display the details of a specific clone, use this command format.

```
smo clone show
-profile profile_name
{-label label_name | -id clone_guid}
```

Example:

```
smo clone show -profile NFSDB_REVENGE -label NFS_C1
```

Output:

```
Clone id: 8abc013111b916e30111b916ffb40001
Clone status: SUCCESS
Clone SID: hsdhc
Clone label: NFS_C1
```

Clone comment: null
Clone start time: 2007-04-03 16:15:50
Clone end time: 2007-04-03 16:18:17
Clone host: hancock
Filesystem: /mnt/weever/data_clone
File: /mnt/weever/data_clone/hsdb/system01.dbf
File: /mnt/weever/data_clone/hsdb/undotbs02.dbf
File: /mnt/weever/data_clone/hsdb/users01.dbf
File: /mnt/weever/data_clone/hsdb/undotbs01.dbf
File: /mnt/weever/data_clone/hsdb/sysaux01.dbf
Backup id: 8abc013111a450480111a45066210001
Backup label: mounttest3
Backup SID: hsdb1
Backup comment:
Backup start time: 2007-03-30 15:26:30
Backup end time: 2007-03-30 15:34:13
Backup host: hancock

Deleting clones

About deleting clones

The clone SID and the clone label are no longer the same in SnapManager 2.1 for Oracle. The label is the unique identifier for each clone in a profile. You can use the clone label or id, but not the clone SID to delete the clone.

Note

Directories specified for certain Oracle parameters in the clone will be destroyed when the clone is deleted, and should only contain data for the cloned database: Archive Log Destinations, Background, Core, and User Dump Destinations.

Deleting clone procedure

To delete a clone, use this command format.

```
smo clone delete
-profile profile_name
[-label clone_label | -id id]
```

Example:

```
smo clone delete -profile NFSDB -label clone1
```

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]
```

Example:

```
smo version
```

Output:

```
SnapManager for Oracle Version: 2.1
```

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. <code>smo_server stop</code>
3	Use the following command to remove the SnapManager software. <code>UninstallSmo</code>
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 system:

- ◆ Go to **Start > All Programs > NetApp > SnapManager for Oracle 2.1.0 > Uninstall SnapManager for Oracle 2.1.0**.
- ◆ Go to **Start > Control Panel**. Double click **Add or Remove Programs** and select **NetApp SnapManager for Oracle 2.1.0**, 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 142

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, 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 SALES1 -label full_backup -verbose
```

Output:

```
SMO-13046 [INFO ]: Operation GUID 8abc013111b9088e0111b908a7560001
starting on Profile SALES1
SMO-08052 [INFO ]: Beginning to connect mount(s) [/mnt/weever/logs,
/mnt/weever/data] from logical snapshot
SMO_SALES1_hsdbs1_F_C_1_8abc013111a450480111a45066210001.
SMO-08025 [INFO ]: Beginning to connect mount /mnt/weever/logs from
snapshot SMO_SALES1_hsdbs1_F_C_1_8abc013111a450480111a45066210001_0
of volume hs_logs.
SMO-08027 [INFO ]: Finished connecting mount /mnt/weever/logs from
snapshot SMO_SALES1_hsdbs1_F_C_1_8abc013111a450480111a45066210001_0
of volume hs_logs.
SMO-08025 [INFO ]: Beginning to connect mount /mnt/weever/data from
snapshot SMO_SALES1_hsdbs1_F_C_1_8abc013111a450480111a45066210001_0
of volume hs_data.
SMO-08027 [INFO ]: Finished connecting mount /mnt/weever/data from
snapshot SMO_SALES1_hsdbs1_F_C_1_8abc013111a450480111a45066210001_0
of volume hs_data.
SMO-08053 [INFO ]: Finished connecting mount(s) [/mnt/weever/logs,
/mnt/weever/data] from logical snapshot
SMO_SALES1_hsdbs1_F_C_1_8abc013111a450480111a45066210001.
SMO-13037 [INFO ]: Successfully completed operation: Backup Mount
SMO-13048 [INFO ]: Operation Status: SUCCESS
SMO-13049 [INFO ]: Elapsed Time: 0:01:00.981
Operation Id [8abc013111b9088e0111b908a7560001] succeeded.
```

```
Operation Id
[8abc013111b909eb01
11b90a02f50001]
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 148
- ◆ [“Gathering troubleshooting information”](#) on page 150
- ◆ [“Troubleshooting upgrade issues”](#) on page 153
- ◆ [“General troubleshooting tips for using the GUI”](#) on page 154
- ◆ [“General troubleshooting tips for working with clones”](#) on page 156
- ◆ [“General troubleshooting tips for SnapManager”](#) on page 157
- ◆ [“Common error conditions”](#) on page 171
- ◆ [“Where to go for more information”](#) on page 173

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, make sure that you can write to the storage system.

- ◆ Is the SnapManager server running?

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

```
smo_server status
```

To stop the SnapManager server, run the command:

```
smo_server stop
```

To start the SnapManager 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 251.

- ◆ 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 in:

- ◆ `/var/log/smo`

Each operation goes into its own log file of the form:

```
smo_operation_factory_id.log
```

- ❖ 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. The `smo.config` file enables you to exclude certain archive log files. The file is located in:
 - `/opt/NTAP/smo/properties/smo.config` (Solaris)
 - `/opt/NetApp/smo/properties/smo.config` (all other platforms)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 compressed log files 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 files include:

- ◆ The steps performed
- ◆ How long they took
- ◆ What the outcome of each step was
- ◆ Any messages 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, send these files to NetApp Global Services for help.

Locating the dump files

The dump result files contain the output for the dump request. The name of the files depend on the information supplied. There is one dump file for the host and one for the server.

- ◆ If you did not specify an operation ID (the GUID or a label), the files are named `smo_dump_results.jar` and `smo_client_dump_results.jar`.

Information included in the dump files

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

The following information is also included in the files:

- ◆ 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
- ◆ SnapManager for Oracle
- ◆ Shared memory maximum
- ◆ Swap space information
- ◆ Memory information
- ◆ Kernel parameters
- ◆ FSTAB

General steps for creating dump files from the CLI

Both the CLI command and the GUI require similar information if you want to create dump files. This section takes you through the basic steps of creating a dump file. For a syntax example, see “[Command information](#)” on page 151.

The basic steps for creating dump files 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 179.

Command information

For details on the options and arguments used in the commands to get dump information, see “[The smo profile dump command](#)” on page 227, the “[The smo operation dump command](#)” on page 219, or the “[The smo system dump command](#)” on page 250.

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 files. Click **Back** to correct any information.

Troubleshooting upgrade issues

Profile name clash when upgrading to SnapManager 2.1 for Oracle

Symptom: The CLI or GUI user gets an error message indicating that there is a profile name clash with another repository.

Explanation: A scenario can arise whereby a profile name that was acceptably unique in SnapManager 2.0 for Oracle is no longer unique within SnapManager 2.1 for Oracle. The problem scenario exists where two profiles:

- ◆ Have the same name
- ◆ Are in different repositories
- ◆ Have respective target DBs on two different servers
- ◆ Need to be referenced by the same OS user

What to do: The workaround is to rename one of the profiles after upgrade.

1. Log on to one of the repositories with a clashing profile name
2. Run: `UPDATE SMO_PROFILE SET NAME = 'NEW_NAME' WHERE NAME = 'OLD_NAME'`

General troubleshooting tips for using the GUI

GUI installation succeeds with errors

Symptom: GUI installation on Windows succeeds with errors.

Explanation: A Windows user installs the SnapManager GUI. The installation succeeds, but you get an error message.

Furthermore, the user does not have a desktop icon to start the SnapManager GUI.

In this scenario, the user account used to install the SnapManager GUI has enough permissions to install SnapManager, but not enough permissions to set up SnapManager icons and shortcuts for all users of the PC. The user account must be able to modify the directory `C:\Documents and Settings\All Users`.

What to do: The user should perform one of the following:

- ◆ Reinstall the SnapManager GUI with different settings. On the "Choose Icon Availability" step, deselect "Make these shortcuts available to all users of this PC". This time the installation should succeed completely.
- ◆ Login using a user account that is not restricted and reinstall 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 password. Once you do this, SnapManager displays the backup list.

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

Performing point in time recovery in GUI displays time only to the minute level

Symptom: A point-in-time recovery in the GUI displays the time in minutes, rather than seconds.

Explanation: When performing a point-in-time recovery with a selected date and time, the user is only allowed to select a date and time to the minute level. Because Oracle allows a selected date and time to the second level, a non-deterministic number of seconds is appended to the user selected date and time.

What to do: If you need accuracy to the nearest second, use the CLI rather than the GUI to recover backups, as it allows you to specify a date and time to the second level. You can also perform point-in-time recovery using an SCN.

Note

This only occurs in the GUI, and only if performing a point-in-time recovery with a date and time.

General troubleshooting tips for working with clones

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 operation abort` command:

```
$ smo operation abort -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 <code>.dbf</code> extension.
2	Repeat the backup operation.
3	Now clone the new backup.

General troubleshooting tips for SnapManager

Command **smo_server start** **fails**

Symptom: The command `smo_server start` fails with message:
SMO-01104: Error invoking command: Multiple exceptions.

Explanation: The most likely cause of this problem is that the SnapManager listening ports (27214 and 27215, by default) are currently in use. In log file, the text "java.net.BindException: Address already in use" appears. This error occurs if another application is currently using the ports. In this case, reconfigure either SnapManager for Oracle or the other application to use different ports. This error can also occur if the `smo_server` is already running, but SnapManager for Oracle did not detect the existing process.

What to do: Reconfigure one of the applications to use different ports. To reconfigure SnapManager, edit the following file:

`/opt/NTAP/smo/properties/smo.config` (Solaris)

`/opt/NetApp/smo/properties/smo.config` (all other platforms)

Locate the following lines and change the listening port numbers to unique values in your environment:

`SMOServer.port=27214`

`SMOServer.rmiRegistry.port=27215`

Then run the following sequence of commands to restart the SnapManager server:

1. `smo_server start`

You will get an error message.

2. `smo_server stop`

This will stop the existing SMO server.

3. `smo_server start`

Now the SMO server should successfully start.

Updating repository results in error message about incomplete operations

Symptom: When you attempt to update a repository that has incomplete operations, you get an error message.

Explanation: Incomplete operations take up space and can interfere with the update. You should always back up the repository before updating. If you have updated the repository and get an error message about incomplete operations, stop the update.

What to do: Use the `smo repository show` command to see which operations are incomplete and abort those operations before updating the repository.

To check which operations are incomplete on the repository before updating it, use the `smo repository show` command.

```
smo repository show -repository -port port -dbname dbname
-host host_name -login -username user1
```

Example:

```
smo repository show -repository -port 1521 -dbname DATAB1
-host server1 -login -username user1
```

If incomplete operations are listed or are currently running on the server, verify with other users that the repository is free to update. If it is, run the `smo operation abort` command with the operation ID and the `-force` option to free resources from incomplete operations before updating the repository.

You can also use the `-force` option to force the old repository schema to be overwritten. However, once you update the repository, you can no longer abort operations performed with a previous version of SnapManager. The resources from the incomplete operations are not freed, but the repository update succeeds.

SnapManager uses default pfile/spfile for Oracle startup

Symptom: When SnapManager restarted the database, it failed.

Your environment is set up so that the `initSID.ora` file is located in a directory other than the `$ORACLE_HOME/dbs` directory. Then you started your Oracle database using `startup pfile`. You put the path to this `initSID.ora` file in this command; however, SnapManager looked in a different location for this file so the database failed to restart.

Explanation: When SnapManager transitions from a Shutdown state into a Mounted or Open state, it uses the default SPFILE or PFILE. If you normally start your database with a specific PFILE or SPFILE by using the `startup pfile=file` syntax, the database restart fails. This is because SnapManager does

not use the file you normally specify when it restarts the database. Instead, SnapManager uses the default location to start the database. Normally, this is `ORACLE_HOME/dbs`.

What to do: If the default `initSID.ora` file is not the correct initialization file for your database, either copy the file to default location (for example, `$ORACLE_HOME/dbs/initSID.ora`) or set up a symbolic link to this location.

Restore fails in SnapMirror configuration

Symptom: The restore operation fails when you attempt to restore a backup of a database volume that was created using SnapMirror.

Explanation: SnapDrive for UNIX has some architectural guidelines for working with mirrored storage systems. Because SnapManager requires SnapDrive for UNIX, these guidelines affect it also.

What to do: Check the SnapDrive for UNIX documentation for information on how it works with SnapMirror and how to handle any issues. In particular, check chapter 1 of the *SnapDrive for UNIX Installation and Administration Guide*.

Space shortages can occur during restore operations

Symptom: SnapManager runs out of space in the `/opt/NetApp/smo/tmp` directory while it is performing a restore operation.

Explanation: SnapManager uses this directory to hold temporary, safe copies while it performs an operation.

What to do: Make sure you allocate sufficient space for this directory. One option is to set it up on a separate file system. If you need additional space, but no more physical disk space is available on the root file system, then create a symbolic link from the `snapmanager_install_dir/tmp` directory to a disk or files system with more space.

Server times out

Symptom: You receive an error message that a time out has occurred waiting for a response from the SnapManager Server.

Explanation: Either the server does not have sufficient memory, or it is unresponsive to commands.

What to do: Increase the `axis.timeout` parameter or free up memory on the SnapManager server or the repository database server.

To increase the `axis.timeout` parameter, edit the following file:

```
/opt/NTAP/smo/properties/smo.config (Solaris)
/opt/NetApp/smo/properties/smo.config (all other platforms)
```

Add the following line:

```
axis.timeout=value
```

Where *value* is the timeout in integer seconds. The default value is 600 (10 minutes).

Restart the SnapManager server to make the new timeout take effect:

```
smo_server restart
```

Profile create operation fails if you use an existing name

Symptom: The profile create operation fails when you try to create a profile with the same name as a profile in a different repository on that system.

Explanation: SnapManager uses the same folder for all profiles. As a result, even though you're using a different repository, you cannot use the name of an existing profile.

What to do: Make sure each profile name on a system is unique regardless of how many repositories you have on that system.

Restore operation fails saying login denied

Symptom: A SnapManager restore operation fails with a message similar to:

```
SMO-13040 [ERROR]: Error while handling failed operation: ORA-01017: invalid username/password; logon denied
```

Explanation: This message indicates that the Oracle database system password changed after the backup occurred. As a result, SnapManager can no longer access the database.

What to do: If you encounter this problem, you need to run the SnapManager `smo operation abort` command to release the resources that were in use when the command failed. Use **one** of the following three scenarios to do this:

- ◆ Change the database password back to the password at the time of the database backup so that SnapManager can access the database. Now run the `smo operation abort` command.
- ◆ Or update the profile associated with that backup and enter the new database password. Now run `smo operation abort`.
- ◆ Or use the `-force` option and run `smo operation abort -force`. This forces SnapManager to clean up after the failed restore operation.

**Backup operation
unable to freeze file
system**

Symptom: SnapManager attempted to back up data files, control files, log files, and archive log files for an ASM database that is spread across multiple volumes. An error occurred, indicating that SnapManager could not freeze the file system.

Explanation: This is a issue with databases spread across multiple volumes that has been seen with versions of Data ONTAP prior to 7.2. In addition, SnapDrive for UNIX has some architectural guidelines for freezing file systems that span multiple storage system volumes.

What to do: Reconfigure your database so that it does not span multiple volumes. It is recommended that you follow the recommendations in technical reports TR 3329 (http://www.netapp.com/tech_library/3329.html) and TR 3349 (http://www.netapp.com/tech_library/3349.html) when laying out your database. You may also want to check the SnapDrive for UNIX documentation for information. In particular, see chapter 1 of the *SnapDrive for UNIX Installation and Administration Guide*.

**The recovery
operation accepts
an SCN above the
maximum**

Symptom: The SnapManager recovery operation completed even though the command used an Oracle System Change Number (SCN) that was above the maximum.

Explanation: Oracle allows you to enter any number SCN except 1. SnapManager is following Oracle's behavior.

**Profile verify
operation fails
because of missing
SID**

Symptom: The profile verify operation fails with a message that SnapManager cannot find the SID in the oratab file. The path is `/var/opt/oracle/oratab` for Solaris and `/etc/oratab` on Linux, AIX and HP-UX.

Explanation: SnapManager always looks for the database SID in the `/etc/oratab` file or the `/var/opt/oracle/oratab` file in order to determine the Oracle home for the database.

What to do: Make sure you include the SID for each database instance in the `/etc/oratab` file. If you manually create the database, make sure you put the SID in the `/etc/oratab` or `/var/opt/oracle/oratab` file.

**Problem occurs
when using RAC
with SnapManager**

Symptom: SnapManager fails when it attempts to perform operations on Real Application Cluster (RAC) database instances.

Explanation: The Oracle sys password and privileges must be the same for all instances of a RAC database that SnapManager handles.

What to do: Check the password on each RAC database instance to make sure that all the instances are using the same password and privileges.

Restore operation fails in ASM configuration

Symptom: A SnapManager restore operation in an ASM configuration fails with the following error:

```
SMO-12018: Error executing RMAN command: [RESTORE DATAFILE
'+DG_EXT1/storasm/datafile/testtab01.dbf' FROM TAG='SMO_storasm_full_1137458676410'];].
The command returned: RMAN-00571:
=====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of restore command at 01/17/2006 09:45:25
RMAN-06026: some targets not found - aborting restore
RMAN-06023: no backup or copy of datafile 7 found to restore
```

Explanation: You are using an RMAN control file for cataloging. You perform a controlfile restore. Because SnapManager uses RMAN to copy ASM files from cloned disk groups into the active disk groups, it requires that the same controlfile used to register a backup be present at the time of the restore operation.

What to do: Use an external database as the RMAN catalog instead of the controlfile. Another option is to restore a controlfile that contains the necessary RMAN information (i.e., one from after the backup you are trying to restore).

SnapManager cannot reconstruct RAW or ASM map

Symptom: SnapManager for Oracle 2.1 fails while attempting to restore, mount, or clone backups taken with previous versions of SnapManager for Oracle on Linux containing RAW or ASM files with error messages similar to:
SMO-11041: Cannot reconstruct "ASM" map for "ORCL:SMO_ASM_RACC_1" on this platform.
SMO-11041: Cannot reconstruct "RAW" map for "/dev/raw2" on this platform.

Explanation: Prior SnapManager for Oracle backups do not contain sufficient information to use ASM or RAW files if the host side storage has changed since the backup was taken. As a precaution, SnapManager 2.1 will not attempt to perform certain operations with older SnapManager for Oracle backups unless additional action is taken.

What to do: Restore, mount, or clone operations may only be performed on the original host. As well, the structural layout of the source database and the Linux raw device map for any datafiles in the backup must not have changed since the backup was taken.

If all of these conditions are met, add the following line to
`/opt/NetApp/smo/properties/smo.config` and retry the operation:

```
services.vhi.allowCloneLinuxCurrentRawMap=true
```

Backup fails for Oracle 10g RAC database when remote instance is in MOUNT or NOMOUNT state

Symptom: The SnapManager backup operation hangs on an Oracle 10g RAC database when one of the remote RAC instances is in the NOMOUNT (startup nomount) or MOUNT (startup mount) state. The Oracle commands alter database begin backup and alter database end backup hang when the database is in this state.

What to do: Shut down the RAC instance that is in the NOMOUNT or MOUNT state. Bug report 4956243 has been filed with Oracle. Oracle resolved this issue with release 10.2.0.3.

Backup of Oracle 10g NFS database fails with RMAN error

Symptom: Backup of an Oracle 10g NFS database fails with an RMAN error similar to:

```
"ORA-27054: NFS file system where the file is created or resides is not mounted with correct options"
```

Explanation: Oracle 10g has a known issue with RMAN registration of datafiles on NFS volumes on some platforms.

What to do: Follow the directions in Oracle Metalink Note 387700.1:

Execute the following SQL statement and restart the target database in SQL*Plus as the sys user:

```
SQL> ALTER SYSTEM SET EVENT='10298 trace name context forever,  
level 32' SCOPE=spfile;
```

```
SQL> SHUTDOWN IMMEDIATE
```

```
SQL> STARTUP
```

Restore and Recover fails

Symptom: When performing a restore and recover operation, SnapManager fails with an error similar to the following:

```
SMO-12014: Error executing SQL command: [RECOVER DATABASE UNTIL  
TIME '2006-01-12:06:37:26'];]. The command returned: ORA-00283:  
recovery session canceled due to errors  
ORA-01610: recovery using the BACKUP CONTROLFILE option must be  
done
```

Explanation: The recover operation must use a backup controlfile. You cannot include the recover option when you perform a restore operation using the controlfiles option.

What to do: Include the `-using-backup-controlfiles` option when performing recovery after a controlfile restore.

controlfile restore fails on 9i RAC setup

Symptom: SnapManager attempts to restore controlfiles for 9i RAC configuration, but the operation fails with the message:

```
SMO-07411 [ERROR]: Oracle RAC srvctl command failed. Command is:  
"srvctl start instance -d <database> -i <instance> -o mount",  
output is "PRKP-1001 : Error starting instance <instance> on node  
<node> ORA-01081: cannot start already-running ORACLE - shut it  
down first"
```

Explanation: Oracle's `srvctl` utility for Oracle 9i does not correctly handle transitioning an instance from a `NOMOUNT` state to a `SHUTDOWN` state (see Metalink SR 5135325.993). SnapManager requires this transition for a controlfile restore if any remote RAC instance is in the `MOUNTED` or `OPEN` state. Without this transition, SnapManager fails in this task. Newer versions of Oracle support this.

What to do: Manually shutdown the remote RAC instances before performing the controlfile restore in SnapManager or install a newer Oracle version that supports the transition of the database instance that SnapManager requires.

Backup fails due to invalid snapshot name

Symptom: You attempt to create or backup a profile containing a colon in the profile name and the operation fails with the message:

```
SMO-08007 [ERROR]: Error creating snapshot. SMO-12111: Error  
executing snapdrive command "/usr/sbin/snapdrive snap create -fs  
<MountPoint> -snapname <SnapshotName> -unrelated": 0001-102 Command  
error: Invalid snapshot name: <SnapshotName>
```

Explanation: SnapManager does not recognize a profile with a colon in the name and fails when you attempt to create or backup such a profile.

What to do: When creating a profile, specify a profile name that does NOT contain a colon.

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 copy` command instead.

RMAN retention policy and SnapManager for Oracle conflicts

Symptom: SnapManager 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 registers with RMAN. RMAN may delete the backup information from its catalog while SnapManager 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 storage system controller 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 username and password failed.

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

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 SQL*Plus 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.

Missing JDBC client driver

Symptom: When verifying a profile, you get an error message similar to:

```
ORACLE_HOME "/mnt/locknessmonster/astoria_9iR2_home" for "fgast1"
is missing the required JDBC client driver. Install the complete
JDBC client driver from Oracle into
"/mnt/locknessmonster/astoria_9iR2_home/jdbc/lib". SMO-13032
[ERROR]: Cannot perform operation: System Verify. Root cause: SMO-
12140: ORACLE_HOME "/mnt/locknessmonster/astoria_9iR2_home" for
"fgast1" is missing the required JDBC client driver. Install the
complete JDBC client driver from Oracle into
"/mnt/locknessmonster/astoria_9iR2_home/jdbc/lib". Operation Id
[N4e0ff04d107c6203ff2b86f0d6797c9f] failed. Error: SMO-12140:
ORACLE_HOME "/mnt/locknessmonster/astoria_9iR2_home" for "fgast1"
is missing the required JDBC client driver. Install the complete
JDBC client driver from Oracle into
"/mnt/locknessmonster/astoria_9iR2_home/jdbc/lib".
```

Explanation: Oracle bundles both the JDBC drivers together. The 10g thin driver is preferable in all cases when using the listener, but the thick driver must match the shared libraries compiled in ORACLE_HOME. The file is named ojdbc14.jar for Oracle 9i and Oracle 10g.

Go to

http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html.

Download the correct JDBC driver for the installed version of Oracle, and install ojdbc14.jar into the directory specified in the error.

Restore of files on NFS volume fails

Symptom: SnapManager fails a restore operation involving files on an NFS volume with an error similar to:

```
SMO-08905 [ERROR]: Internal error: Snapdrive snap-restored file
"/u/oracle/controlfiles/SMOBakCtl_1173235929636_2" not present in
active filesystem.
```

Explanation: This is caused by the operating system caching old file system data after a snap restore operation is performed on the storage system.

What to do: Ensure the file system is mounted with the recommended options specified in NetApp KB Article 7518 available at:

<http://now.netapp.com/Knowledgebase/solutionarea.asp?id=kb7518>

Restore controlfiles fails if controlfiles are multiplexed onto ASM storage AND on non-ASM storage

Symptom: If a database has its controlfiles multiplexed onto ASM storage and on non-ASM storage, SnapManager for Oracle will fail to restore the controlfiles with an error message similar to the following:

```
SMO-03024: Unable to restore controlfiles multiplexed on ASM and
non-ASM storage. Manual restoration can be attempted by mounting
the backup and copying a controlfile to all controlfile locations.
The controlfiles stored on ASM storage are [...]. The controlfiles
stored on non-ASM storage are [...].
```

When a backup is taken with SnapManager 2.1 for Oracle, the following is displayed to warn about this situation.

```
SMO-02028: Database abc contains multiplexed controlfiles on ASM
storage and non-ASM storage. smo will not be able to restore
controlfiles automatically. The controlfiles stored on ASM storage
are [...]. The controlfiles stored on non-ASM storage are [...].
```

SnapManager for Oracle versions prior to 2.1 did not have this warning.

What to do: To restore the controlfiles:

1. Mount the backup using the `smo backup mount` command.
2. Copy a controlfile to all of the database controlfile locations.

You must copy the same controlfile to all of the locations so that all controlfiles are in sync when the database instance is opened, or you will receive an ORA-00214 error.

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 SnapManager server administration.
API	18000-18999	Messages associated with the SnapManager application program interface.
AUTH	20000-20999	Messages associated with credential authorization.

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 49.

- ◆ *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 database information
See the readme files and other documentation that you received with your host operating system and database software.

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 177
- ◆ “[Common options for handling messages shown on the console](#)” on page 179
- ◆ “[The smo backup create command](#)” on page 180
- ◆ “[The smo backup delete command](#)” on page 184
- ◆ “[The smo backup list command](#)” on page 186
- ◆ “[The smo backup mount command](#)” on page 188
- ◆ “[The smo backup restore command](#)” on page 190
- ◆ “[The smo backup show command](#)” on page 193
- ◆ “[The smo backup update command](#)” on page 196
- ◆ “[The smo backup unmount command](#)” on page 198
- ◆ “[The smo backup verify command](#)” on page 200
- ◆ “[The smo clone create command](#)” on page 202
- ◆ “[The smo clone delete command](#)” on page 204
- ◆ “[The smo clone list command](#)” on page 205
- ◆ “[The smo clone show command](#)” on page 206
- ◆ “[The smo clone template command](#)” on page 208
- ◆ “[The smo cmdfile command](#)” on page 210
- ◆ “[The smo credential clear command](#)” on page 212
- ◆ “[The smo credential list command](#)” on page 213
- ◆ “[The smo credential set command](#)” on page 214
- ◆ “[The smo help command](#)” on page 216
- ◆ “[The smo operation abort command](#)” on page 217
- ◆ “[The smo operation dump command](#)” on page 219
- ◆ “[The smo profile create command](#)” on page 221
- ◆ “[The smo profile delete command](#)” on page 226
- ◆ “[The smo profile dump command](#)” on page 227
- ◆ “[The smo profile list command](#)” on page 228
- ◆ “[The smo profile show command](#)” on page 229
- ◆ “[The smo profile sync command](#)” on page 231
- ◆ “[The smo profile update command](#)” on page 233

- ◆ “The smo profile verify command” on page 237
- ◆ “The smo repository create command” on page 238
- ◆ “The smo repository delete command” on page 240
- ◆ “The smo repository show command” on page 242
- ◆ “The smo repository update command” on page 244
- ◆ “The smo_server restart command” on page 246
- ◆ “The smo_server start command” on page 247
- ◆ “The smo_server status command” on page 248
- ◆ “The smo_server stop command” on page 249
- ◆ “The smo system dump command” on page 250
- ◆ “The smo system verify command” on page 251
- ◆ “The smo version command” on page 252

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 with the `-verbose` option.
- ◆ 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 210.
- ◆ 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 179.

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 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 creates 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 hexadecimal string generated by the system. (For more information about the GUID, see [“General notes about the commands”](#) on page 177.)

Syntax:

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

Note

Before you run 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 221.

Options:

-profile *name*

Specifies the database you want to back up. The profile contains the identifier of the database and other database information.

-auto

Automatically selects the type of backup based on the state of the database:

- ◆ 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 `-force` option is specified and none of the above are true, it does an offline backup.

-online

- ◆ Specifies an online database backup.

You can take an online backup of a RAC database, as long as the primary instance is MOUNTED and any remote instance is OPEN. Use `-force` for online backups if the local instance is SHUTDOWN, or no instance is OPEN. If the local instance is SHUTDOWN and at least one instance is OPEN, then using `-force` changes the local instance to MOUNTED. If no instance is OPEN, using `-force` changes the local instance to OPEN.

Note

The version of Oracle must be 10.2.0.3 or higher or the database will hang if any instance in the RAC is mounted.

-offline

Specifies an offline backup while the database is shut down. The database is mounted, but not open. If the database is up when you specify this option, the backup fails. Use the `-force` option to have SnapManager attempt to alter the database state before making the backup.

-full

Specifies that SnapManager makes a backup of the entire database. This includes all the data, and log and control files.

-files *list*

Specifies that SnapManager backs up only the specified database files plus the log and control files. Separate file names with spaces. If the database is up, SnapManager ensures that the appropriate tablespaces are in online backup mode.

-tablespaces *list*

Specifies that SnapManager backs up only the specified database tablespaces plus the log and control files. Separate tablespace names with spaces. If the database is up, SnapManager ensures that the appropriate tablespaces are in online backup mode.

-label *name*

Specifies an optional name for this backup. This name must be unique within the profile. The name can contain letters, numbers, underscore (`_`), and hyphen (`-`). It cannot start with a hyphen.

If you do not specify 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. SnapManager uses a 24-hour clock.

For example, if you performed a full backup with the database offline on Jan. 16, 2007, at 5:45:16 p.m. Eastern standard time, SnapManager would create the label F_C_20070116174516EST.

-comment *string*

Specifies an optional comment to describe this backup. Enclose the string in single quotes (').

Note

Some shells strip off quote marks. If that is true for your shell, you must escape the quote with a backslash (\). For example, you might need to enter: \ ' this is a comment \ '.

-retain-forever

Specifies that SnapManager make the backup ineligible for deletion by the retention policy.

-retain-normal

Specifies that SnapManager make the backup eligible for the normal retention policy.

-verify

Specifies to verify that the files in the backup are not corrupt by running the Oracle dbv utility. For information about separately verifying a backup, see the [“Using the smo backup verify command”](#) on page 200.

Note

If you specify the -verify option, the backup operation does not complete until the verify operation completes.

-force

Specifies that SnapManager force a state change if the database is not in the correct state. For example, 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.

With an online RAC database backup, use `-force` if the local instance is SHUTDOWN, or no instance is OPEN. If the local instance is SHUTDOWN and at least one instance is OPEN, then using `-force` changes the local instance to MOUNTED. If no instance is OPEN, using `-force` changes the local instance to OPEN.

Note

The version of Oracle must be 10.2.0.3 or higher or the database will hang if any instance in the RAC is mounted.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

**Example smo
backup create
command**

The following example creates a full offline backup.

```
smo backup create -profile SALES1 -full -offline  
-label full_backup_sales_May -force
```

Output:

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

The smo backup delete command

Using the smo backup delete command

The `smo backup delete` command enables you remove backups. Use this command 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 221 and “[The smo profile update command](#)” on page 233.)

Syntax:

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

Options:

-profile *profile_name*

Specifies the database associated with the backup you want to remove. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to delete the backup with the specified GUID. The GUID is generated by SnapManager when you create a backup. Use the `smo backup list` command to display the GUID for each backup.

-label *label_name*

Specifies to delete the backup with the specified label.

-retain *number*

Specifies to delete successful backups until you have only the specified number remaining. The oldest backups are deleted first.

For example, if you had five backups from May, June, July, August, and September and you specified `-retain 3`, SnapManager would delete the backups from May and June.

-all

Specifies to delete all backups.

-force

Specifies that SnapManager force the removal of the backup. 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 `-force` on the command line lets SnapManager delete the backup even though it cannot connect with RMAN.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

**Example smo
backup delete
command**

The following example deletes the backup with the label `full_backup_sales_May`.

```
smo backup delete -profile SALES1 -label full_backup_sales_May
```

Output:

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

The smo backup list command

Using the smo backup list command

The `smo backup list` command displays information about all of the backups in a profile.

Syntax:

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

-profile *profile_name*

Specifies the database you want to list backups for. The profile contains the identifier of the database and other database information.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Includes additional information about the backup in the output. Also specifies that error, warning, and informational messages are displayed on the console.

Example smo backup list command

The following example command lists the backups for the SALES1 profile.

```
smo backup list -profile SALES1
```

Output:

Start Date	Label	Status	Scope	Mode
-----	-----	-----	-----	-----
2007-03-20 21:12:15	backup13	SUCCESS	PARTIAL	OFFLINE
2007-03-20 21:10:11	fail2	FAILED	PARTIAL	ONLINE
2007-03-20 21:06:22	fail1	FAILED	PARTIAL	OFFLINE
2007-03-20 21:04:27	backup12	SUCCESS	FULL	ONLINE
2007-03-16 18:54:02	backup11	SUCCESS	FULL	ONLINE
2007-03-15 21:47:11	backup8	SUCCESS	FULL	ONLINE
2007-03-15 21:38:43	label5	SUCCESS	FULL	ONLINE
2007-03-15 20:41:19	label2	FAILED	FULL	OFFLINE
2007-03-15 20:39:55	label1	SUCCESS	FULL	ONLINE

The following example command lists the backups for the SALES1 profile with the addition of the retention policy, the ID, and any comments entered about the backup.

```
smo backup list -profile SALES1 -verbose
```

Start Date	Label	Status	Scope	Mode	Retention	ID	Comment
-----	-----	-----	-----	----	-----	-----	-----
2007-03-20 21:12:15	bkup13	SUCCESS	PARTIAL	OFFLINE	NORMAL	ff8...30001	test comment
2007-03-20 21:10:11	fail2	FAILED	PARTIAL	ONLINE	N/A	ff8...20001	
2007-03-20 21:06:22	fail1	FAILED	PARTIAL	OFFLINE	N/A	ff8...70001	
2007-03-20 21:04:27	bkup12	SUCCESS	FULL	ONLINE	NORMAL	ff8...270001	
2007-03-16 18:54:02	bkup11	SUCCESS	FULL	ONLINE	NORMAL	ff8...d20001	
2007-03-15 21:47:11	bkup8	SUCCESS	FULL	ONLINE	NORMAL	ff8...e0001	
2007-03-15 21:38:43	label5	SUCCESS	FULL	ONLINE	NORMAL	ff8...50001	
2007-03-15 21:20:40	label4	FAILED	FULL	OFFLINE	N/A	ff8...a0001	
2007-03-15 20:44:55	bkup3	FAILED	FULL	OFFLINE	N/A	ff8...820001	
2007-03-15 20:41:19	label2	FAILED	FULL	OFFLINE	N/A	ff8...80001	
2007-03-15 20:39:55	label1	SUCCESS	FULL	ONLINE	NORMAL	ff8...10001	

The smo backup mount command

Using the smo backup mount command

The `smo backup mount` command mounts 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.

Syntax:

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

Options:

-profile *profile_name*

Specifies the database you want to mount a backup of. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to mount the backup with the specified GUID. The GUID is generated by SnapManager when you create a backup. Use the `smo backup list` command to display the GUID for each backup.

-label *label_name*

Specifies to mount the backup with the specified label.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo backup mount command

The following example command mounts the backup with the label `full_backup_sales_May`.

```
smo backup mount -profile SALES1 -label full_backup_sales_May  
-verbose
```

Output:

```
SMO-13046 [INFO ]: Operation GUID 8abc013111b9088e0111b908a7560001  
starting on Profile SALES1  
SMO-08052 [INFO ]: Beginning to connect mount(s) [/mnt/weever/logs,  
/mnt/weever/data] from logical snapshot  
SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001.  
SMO-08025 [INFO ]: Beginning to connect mount /mnt/weever/logs from  
snapshot SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001_0  
of volume hs_logs.  
SMO-08027 [INFO ]: Finished connecting mount /mnt/weever/logs from  
snapshot SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001_0  
of volume hs_logs.  
SMO-08025 [INFO ]: Beginning to connect mount /mnt/weever/data from  
snapshot SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001_0  
of volume hs_data.  
SMO-08027 [INFO ]: Finished connecting mount /mnt/weever/data from  
snapshot SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001_0  
of volume hs_data.  
SMO-08053 [INFO ]: Finished connecting mount(s) [/mnt/weever/logs,  
/mnt/weever/data] from logical snapshot  
SMO_SALES1_hbdb1_F_C_1_8abc013111a450480111a45066210001.  
SMO-13037 [INFO ]: Successfully completed operation: Backup Mount  
SMO-13048 [INFO ]: Operation Status: SUCCESS  
SMO-13049 [INFO ]: Elapsed Time: 0:01:00.981  
Operation Id [8abc013111b9088e0111b908a7560001] succeeded.
```

The smo backup restore command

Using the smo backup restore command

The `smo backup restore` command restores backups of a database or a portion of a database and then optionally recovers the database information.

Syntax:

```
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-controlfiles]]
  [-force]
  [-quiet | -verbose]
```

Options:

-profile *profile_name*

Specifies the database you want to restore. The profile contains the identifier of the database and other database information.

-label *label_name*

Specifies to restore the backup with the specified label.

-id *guid*

Specifies to restore the backup with the specified GUID. The GUID is generated by SnapManager when you create a backup. Use the `smo backup list -verbose` command to display the GUID for each backup.

-complete

Specifies to restore everything in the backup.

-tablespaces *list*

Specifies to restore only the specified tablespaces from the backup. Use spaces to separate the names in the list.

-files *list*

Specifies to restore only the specified files from the backup. Use spaces to separate the names in the list. If database is up, SnapManager ensures that the tablespace containing the files is offline.

-controlfiles

Specifies to restore only the control files. You cannot specify `-controlfiles` with `-complete`.

-recover

Specifies that SnapManager recover the database after restoring it. You must also specify the point to which you want SnapManager to recover the database using one of the following options:

-nologs

Specifies that SnapManager recover the database to the time of the backup.

-alllogs

Specifies that SnapManager recover the database using all available logs.

-until date

Specifies that SnapManager recover the database up to the date and time specified. Use the format year:month:date:hour:minute:second (*yyyy-mm-dd:HH:MM:SS*). Note that this is the database time, not necessarily the time on the your workstation. The hours may use either 12- or 24-hour format, depending on the database setting.

-until scn

Specifies that SnapManager rolls forward the Oracle SCN log files until it reaches the specified SCN number.

-using-backup-controlfile

Specifies that SnapManager restore using the backup control file.

-force

Specifies that SnapManager change the database state to a lower state than its current state. For RAC, include the `-force` 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.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo backup restore command

The following example restores one database file and recovers the database using all available logs.

```
smo backup restore -profile SALES1 -label full_backup_sales_May  
-files /mnt/server/datafiles/users01.dbf  
-recover -alllogs
```

Output:

Operation Id [8abc01ec0e78f3e2010e78f3fdd00001] succeeded.

The smo backup show command

Using the smo backup show command

The `smo backup show` command displays detailed information about a backup.

Syntax:

```
smo backup show
  -profile profile_name
  [-label backup_label | -id backup_guid]
  [-quiet | -verbose]
```

-profile *profile_name*

Specifies the database for which to show backups. The profile contains the identifier of the database and other database information.

-label *backup_label*

Specifies the name of the backup.

-id *backup_guid*

Specifies the backup ID.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console, as well as any clone and verification information.

Example smo backup show command

The following example command shows detailed information about the backup.

```
smo backup show -profile SALES1 -label BTNFS
```

Output:

```
Backup id: 8abc013111a450480111a45066210001
Backup status: SUCCESS
Backup type: FULL
Backup mode: OFFLINE
Mount status: NOT MOUNTED
Backup label: BTNFS
Backup comment:
```

```

RMAN Tag: SMO_BTNFS_1175283108815
Backup start time: 2007-03-30 15:26:30
Backup end time: 2007-03-30 15:34:13

Verification status: OK
Backup Retention Policy: NORMAL
Backup database: hsd1
  SCN: 2700620
  Tablespace: SYSAUX
    Datafile: /mnt/weever/data/hsdb/sysaux01.dbf [ONLINE]
  ...
Host: hancock
  Filesystem: /mnt/weever/data
    File: /mnt/weever/data/hsdb/SMOBakCtl_1175283005231_0
  ...
Volume: hs_data
  Snapshot: SMO_HSDBR_hsd1_F_C_1_
    8abc013111a450480111a45066210001_0
    File: /mnt/weever/data/hsdb/SMOBakCtl_1175283005231_0
  ...

```

If you add the `-verbose` option to this command, in addition to the previous output, you will get the following additional information.

```

...
CLONES:
  Clone id: 8abc013111b916e30111b916ffb40001
  Clone status: SUCCESS
  Clone SID: hsd1c
  Clone label: hsd1c
  Clone start time: 2007-04-03 16:15:50
  Clone end time: 2007-04-03 16:18:17
  Host: hancock
    Filesystem: /mnt/weever/data_clone
    File: /mnt/weever/data_clone/hsdb/users01.dbf
  ...
VERIFIES:
  Verify id: 8abc013111b919c10111b919da700001
  Backup status: SUCCESS
  Backup start time: 2007-04-03 16:18:58
  Backup end time: 2007-04-03 16:21:12
  Verify result: OK
  Verify output: DBVERIFY - Verification starting : FILE =
/u/user1/p4/capstan/2.1/mnt/_mnt_weever_data_SMO_HSDBR_hsd1_F_C_1_
_8abc013111a450480111a45066210001/hsdb/sysaux01.dbf
  ...
DBVERIFY - Verification complete

```

```
Total Pages Examined           : 49920
Total Pages Processed (Data)    : 11669
Total Pages Failing (Data)      : 0
Total Pages Processed (Index)   : 10279
Total Pages Failing (Index)     : 0
Total Pages Processed (Other)   : 16180
Total Pages Processed (Seg)     : 0
Total Pages Failing (Seg)       : 0
Total Pages Empty               : 11792
Total Pages Marked Corrupt      : 0
Total Pages Influx              : 0
Highest block SCN               : 2700634 (0.2700634)
...
```

The smo backup update command

Using the smo backup update command

The `smo backup update` command sets the retention policy for an existing backup.

Syntax:

```
smo backup update
  -profile profile_name
  {-id guid | -label label_name}
  {-retain-forever | -retain-normal}
  [-quiet | -verbose]
```

Options:

-profile *profile_name*

Specifies the database you want to update a backup of. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to verify the backup with the specified GUID. The GUID is generated by SnapManager when you create a backup. Use the `smo backup list` command to display the GUID for each backup.

-label *label_name*

Specifies to verify the backup with the specified label.

-retain-forever

Specifies that SnapManager make the backup ineligible for deletion by the retention policy.

-retain-normal

Specifies that SnapManager make the backup eligible for the normal retention policy.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

**Example smo
backup update
command**

The following example updates the backup labeled full_backup_sales_May to never be automatically deleted.

```
smo backup update -profile SALES1 -label full_backup_sales_May  
-retain-forever
```

The smo backup unmount command

Using the smo backup unmount command

The `smo backup unmount` command unmounts a backup. You should only need to use this command if you are using an external tool such as RMAN. SnapManager automatically unmounts 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.

Syntax:

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

Options:

-profile *profile_name*

Specifies the database you want to unmount a backup of. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to unmount the backup with the specified GUID. The GUID is generated by SnapManager when you create a backup. Use the `smo backup list` command to display the GUID for each backup.

-label *label_name*

Specifies to unmount the backup with the specified label.

-force

Specifies that SnapManager unmount the backup even if there are problems freeing the resources associated with the backup. SnapManager 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 are errors in the log.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo backup unmount command

The following is an example of an unmount operation:

```
# smo backup unmount -label test -profile SALES1 -verbose
```

Output:

```
SMO-13046 [INFO ]: Operation GUID 8abc013111b909eb0111b90a02f50001
starting on Profile SALES1
SMO-08028 [INFO ]: Beginning to disconnect connected mount(s)
[/u/user1/p4/capstan/2.1/mnt/_mnt_weever_logs_SMO_SALES1_hbdb1_F_C_
_1_8abc013111a450480111a45066210001,
/u/user1/p4/capstan/2.1/mnt/_mnt_weever_data_SMO_SALES1_hbdb1_F_C_
_1_8abc013111a450480111a45066210001].
SMO-08030 [INFO ]: Done disconnecting connected mount(s)
[/u/user1/p4/capstan/2.1/mnt/_mnt_weever_logs_SMO_SALES1_hbdb1_F_C_
_1_8abc013111a450480111a45066210001,
/u/user1/p4/capstan/2.1/mnt/_mnt_weever_data_SMO_SALES1_hbdb1_F_C_
_1_8abc013111a450480111a45066210001].
SMO-13037 [INFO ]: Successfully completed operation: Backup Unmount
SMO-13048 [INFO ]: Operation Status: SUCCESS
SMO-13049 [INFO ]: Elapsed Time: 0:00:33.715
Operation Id [8abc013111b909eb0111b90a02f50001] succeeded.
```

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

Syntax:

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

Options:

-profile *profile_name*

Specifies the database you want to verify a backup of. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to verify the backup with the specified GUID. The GUID is generated by SnapManager when you create a backup. Use the `smo backup list` command to display the GUID for each backup.

-label *label_name*

Specifies to verify the backup with the specified label.

-force

Specifies that SnapManager force the database into the necessary state to perform the verify operation.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

**Example smo
backup verify
command**

The following example command verifies the backup with the label
full_backup_sales_May.

```
smo backup verify -profile SALES1 -label full_backup_sales_May  
-quiet
```

Output:

```
DBVERIFY - Verification starting : FILE =  
+SMO_1_1161675083835/smoasm/datafile/data.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 clone a full backup.

Syntax:

```
smo clone create
  -profile profile_name
  {-backup-id backup_guid | -backup-label backup_label_name |
  -current}
  -newsid new_sid
  [-host target_host]
  -label clone_label
  [-comment string]
  -clonespec full_path_to_clonespec_file
  [-quiet | -verbose]
```

Options:

-profile *profile_name*

Specifies the database you want to clone. The profile contains the identifier of the database and other database information.

-backup-id *guid*

Specifies to clone the backup with the specified GUID. The GUID is generated by SnapManager when you create a backup. Use the `smo backup list -verbose` command to display the GUID for each backup.

-backup-label *label_name*

Specifies to clone the backup with the specified label.

-current

Specifies to backup and clone from the current state of the database.

-newsid *new_sid*

Specifies a new, unique Oracle SID for the cloned database. The SID value is a maximum of eight characters. Oracle does not allow running two databases with the same SID on the same host simultaneously.

-host *target_host*

Specifies the name of the host to create the clone on (if cloning to a different host than the host on which the database resides). SnapManager for Oracle must be running on the target host, and the user must have credentials to access the target host.

-label *clone_label*

Specifies a label for the clone.

-comment *string*

Specifies an optional comment to describe this clone. Enclose the string in single quotes (').

Note

Some shells strip off quote marks. If that is true for your shell, you must escape the quote with a backslash (\). For example, you might need to enter: `\' this is a comment\'`.

-clonespec *full_path_to_clonespec_file*

Specifies the path to the clone specification file. This can be a relative or absolute pathname. For information about creating a clone specification, refer to [“Creating a clone specification”](#) on page 95.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console, as well as any comments about the clone.

Example smo clone create command

The following example command clones the backup with the label `full_backup_sales_May` using a clone specification created for this clone. Once the `smo clone create` command completes, the new database is up and running.

```
smo clone create -profile SALES1 -backup-label
full_backup_sales_May -newsid CLONE -label sales1_clone -clonespec
/opt/NetApp/smo/clonespecs/sales1_clonespec.xml
```

Output:

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

The smo clone delete command

Using the smo clone delete command

The `smo clone delete` command deletes the specified clone of a database.

The clone SID and the clone label are no longer the same in SnapManager 2.1 for Oracle. The label is the unique identifier for each clone in a profile. You can use the clone label or id, but not the clone SID to delete the clone.

Syntax:

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

Options:

-profile *profile_name*

Specifies the database you want to delete a clone of. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to delete the clone with the specified GUID. The GUID is generated by SnapManager when you create a clone. Use the `smo clone list` command to display the GUID for each clone.

-label *label_name*

Specifies to delete the clone with the specified label.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo clone delete command

The following example deletes the clone labeled SALES_May.

```
smo clone delete -profile SALES1 -label SALES_May
```

The smo clone list command

Using the smo clone list command

The `smo clone list` command lists the clones of the database for a given profile.

Syntax:

```
smo clone list
  -profile profile_name [-force]
  [-quiet | -verbose]
```

Options:

-profile *profile_name*

Specifies the database you want to list the clones of. The profile contains the identifier of the database and other database information.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo clone list command

The following example lists the database clones in the SALES1 profile. The ID and Label fields are truncated in the example to better fit the page.

```
smo clone list -profile SALES1 -verbose
```

Output:

ID	Status	SID	Host	Label	Comment
-----	-----	-----	-----	-----	-----
8ab...01	SUCCESS	hsdbc	server1	back1clone	

The smo clone show command

Using the smo clone show command

The `smo clone show` command displays information about the database clones for the specified profile.

Syntax:

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

Options:

-profile *profile_name*

Specifies the database you want to list the clones of. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to show information about the clone with the specified GUID. The GUID is generated by SnapManager when you create a clone. Use the `smo clone list` command to display the GUID for each clone.

-label *label_name*

Specifies to show information about the clone with the specified label.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo clone show command

The following example displays information about the clone labeled `full_backup_sales_May`.

```
smo clone show -profile SALES1 -label full_backup_sales_May
-verbose
```

Output:

```
Clone id: 8abc013111b916e30111b916ffb40001
Clone status: SUCCESS
Clone SID: hsdbsc
Clone label: hsdbsc
Clone comment: null
Clone start time: 2007-04-03 16:15:50
Clone end time: 2007-04-03 16:18:17
Clone host: hancock
  Filesystem: /mnt/weever/data_clone
    File: /mnt/weever/data_clone/hsdb/sysaux01.dbf
    File: /mnt/weever/data_clone/hsdb/undotbs01.dbf
    File: /mnt/weever/data_clone/hsdb/users01.dbf
    File: /mnt/weever/data_clone/hsdb/system01.dbf
    File: /mnt/weever/data_clone/hsdb/undotbs02.dbf
Backup id: 8abc013111a450480111a45066210001
Backup label: full_backup_sales_May
Backup SID: hsdbs1
Backup comment:
Backup start time: 2007-03-30 15:26:30
Backup end time: 2007-03-30 15:34:13
Backup host: server1
```

The smo clone template command

Using the smo clone template command

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

Syntax:

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

Options:

-profile *profile_name*

Specifies the database you want to create a clone specification of. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to create a clone specification from the backup with the specified GUID. The GUID is generated by SnapManager when you create a backup. Use the `smo backup list` command to display the GUID for each backup.

-label *label_name*

Specifies to create a clone specification from the backup with the specified label.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo clone template command

The following example command creates a clone specification template from the backup with the label `full_backup_sales_May`. 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 96.

```
smo clone template -profile SALES1  
-label full_backup_sales_May
```

The smo cmdfile command

Using the smo cmdfile command

Use the `smo cmdfile` command if the shell on your host limits the number of characters that can appear on a command line. Put a single command in a text file and use `smo cmdfile` command to execute the command in the text file.

Note

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

You can put only one `smo` command in a text file. Within the text file, drop the `smo` from the command syntax.

Syntax:

```
smo cmdfile
  -file file_name
  [-quiet | -verbose]
```

Options:

-file *file_name*

Specifies the path to text file containing the command you want to execute.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo cmdfile command

This example creates a profile by putting the `profile` command in a text file and then calling the `smo cmdfile` command.

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

```
profile create -profile SALES1 -repository -dbname SNAPMGRR  
-login -username server1_user -password ontap -port 1521 -host  
server1 -database -dbname SMOASM -sid SMOASM -login -username sys  
-password oracle -port 1521 -host bismarck -osaccount oracle  
-osgroup db2
```

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

```
smo cmdfile -file /tmp/command.txt
```

The smo credential clear command

Using the smo credential clear command

To clear all of the credentials for a user, use this command.

```
smo credential clear  
[-quiet | -verbose]
```

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo credential clear command

Example:

This example clears all of the credentials for the user running the command.

```
smo credential clear -verbose
```

Output:

```
SMO-20024 [INFO ]: Cleared credentials for user "user1".
```

The smo credential list command

Using the smo credential list command

To list credentials for an OS user, use this command.

```
smo credential list
    [-quiet | -verbose]
```

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo credential list command

Example:

This example displays all of the credentials for the user running the command.

```
smo credential list
```

Output:

```
Credential cache for OS user "user1":
```

```
Repositories:
```

```
    hancock_smoke_user@SMOREPO/hotspur:1521
    bismarck_smoke_user@SMOREPO/hotspur:1521
    user1_1@SMOREPO/hotspur:1521
```

```
Profiles:
```

```
    HSDBR (Repository: user1_2_1@SMOREPO/hotspur:1521)
    PBCASM (Repository: user1_2_1@SMOREPO/hotspur:1521)
    HSDB (Repository: hancock_smoke_user@SMOREPO/hotspur:1521)
```

```
[PASSWORD NOT SET]
```

```
Hosts:
```

```
    bismarck
    saratoga
    revenge
    hancock
```

The smo credential set command

Using the smo credential set command

The `smo credential set` command lets you set the credentials for users to access secure resources, such as hosts, repositories and databases.

Syntax:

```
smo credential set
  {-host
    -name host_name } |
    -password host_password
  {-profile
    -name profile_name
    -password profile_password } |
  {-repository
    -dbname repo_dbname
    -host repo_host
    -login
      -username repo_username
      -password repo_password
    -port repo_port }
  [-quiet | -verbose]
```

Options:

-host

Specifies credentials for a host computer.

-name *host_name*

Specifies the host name of the computer.

-password *host_password*

Specifies the password used to access the host computer.

-profile

Specifies credentials for a profile.

-name *profile_name*

Specifies the profile with which the database is associated.

-password *profile_password*

Specifies the profile with which the database is associated.

-repository

Specifies credentials for a repository.

-dbname *repo_dbname*

Specifies the name of the database that stores the profile. Use either the global name or the SID.

-host *repo_host*

Specifies the name or IP address of the host computer the repository database runs on.

-login

Starts the repository login details.

-username *repo_username*

Specifies the user name needed to access the database that stores the repository.

-password *repo_password*

Specifies the password needed to access the database that stores the repository.

-port *repo_port*

Specifies the TCP port number used to access the database that stores the repository.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo credential set command

The following example set credentials for a repository.

```
smo credential set -repository -host bismarck -dbname repol  
-password simba
```

Output:

```
SMO-12345 [INFO ]: Updating credential cache for OS user "admin1"  
SMO-12345 [INFO ]: Set repository credential for user "user1" on  
repol@bismarck.  
Operation Id [Nff8080810da9018f010da901a0170001] succeeded.
```

For more information about setting credentials, refer to [“Allowing access to SnapManager secured resources”](#) on page 57.

The smo help command

Using the smo help command

The `smo help` command displays information about the SnapManager commands and their options. If you do not supply a command name, it displays a list of valid commands. If you supply a command name, it displays the syntax for that command.

Syntax:

```
smo help [command_name]  
        [-quiet | -verbose]
```

Options:

command_name

The command name can be:

- ◆ backup
- ◆ clone
- ◆ cmdfile
- ◆ credential
- ◆ help
- ◆ operation
- ◆ profile
- ◆ repository
- ◆ system
- ◆ version

Example smo help command

The following example command displays information about the `smo operation` command.

```
smo help operation
```

Output:

```
Usage syntax for "smo operation":  
smo operation abort -profile <profile> -label <label> -id <id>  
-force  
smo operation dump -profile <profile> -label <label> -id <id>
```

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 exception occurs that leaves the operation in a permanently incomplete state, 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 `smo operation abort` command to undo the action.

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

Syntax:

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

Options:

-profile *profile_name*

Specifies the database for which you want to stop or undo an operation. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to stop the current operation, or undo the most recent operation, on the operation with the specified GUID. The GUID is generated by SnapManager when you the operation begins.

-label *label_name*

Specifies to stop the current operation, or undo the most recent operation, on the operation with the specified label.

-force

Specifies that SnapManager 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.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

**Example smo
operation abort
command**

The following example command stops the current operation on the backup with the GUID 8abc01ec0e78f3e2010e78f3fdd00001.

```
smo operation abort -profile SALES1  
-id 8abc01ec0e78f3e2010e78f3fdd00001
```

The smo operation dump command

Using the smo operation dump command

The `smo operation dump` command creates two dump files containing information about an operation. There is a client dump file and a server dump file. These files can be helpful if you need to troubleshoot a problem with the operation.

Syntax:

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

Options:

-profile *profile_name*

Specifies the database for which you want to create the dump files. The profile contains the identifier of the database and other database information.

-id *guid*

Specifies to create dump files for the operation with the specified GUID. The GUID is generated by SnapManager when the operation begins.

-label *label_name*

Specifies to create dump files for the operation.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo operation dump command

The following example command creates dump files for the backup with the GUID 8abc01ec0e78f3e2010e78f3fdd00001.

```
smo operation dump -profile SALES1
-id 8abc01ec0e78f3e2010e78f3fdd00001
```

Output:

```
SMO-15035: Client dump file is located at  
"server.domain.com:/u/admin/.netapp/smo/2.1.0/admin_smo_client_dump_results.jar".  
Server dump file is located at  
"server.domain.com:/tmp/smo_dump_results.jar".
```

The smo profile create command

Using the smo profile create command

The `smo profile create` command creates a profile of a database in a repository. This profile contains the host and database credentials SnapManager needs in order to perform its backup and restore operations.

To update an existing profile, use the command `smo profile update` instead of `smo profile create`.

Syntax:

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

Options:

-profile *profile_name*

Specifies the name of the profile. This name can be up to 30 characters long and must be unique within the host.

-profile-password *profile_password*

Specify the password for the profile.

-repository

The options that follow **-repository** specify the details of the database that stores the profile:

-dbname *repo_dbname*

Specifies the name of the database that stores the profile. Use either the global name or the SID.

-host *repo_host*

Specifies the name or IP address of the host computer the repository database runs on.

-login

Starts the repository login details.

-username *repo_username*

Specifies the user name needed to access the database that stores the repository.

-port *repo_port*

Specifies the TCP port number used to access the database that stores the repository.

-database

The options that follow **-database** specify the details of the database that the profile describes. This is the database that will be backed up, restored, and so on.

-dbname *db_dbname*

Specifies the name of the database that the profile describes. Use either the global name or the SID.

-host *db_host*

Specifies the name or IP address of the host computer the database runs on.

-login

Starts the database login details.

-username *db_username*

Specifies the user name needed to access the database that the profile describes.

-password *db_password*

Specifies the password needed to access the database that the profile describes.

-sid *db_sid*

Specifies the SID of the database that the profile describes. 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 `-sid` option. For example, if you are using Oracle RAC, you must specify the SID of the RAC instance on the RAC node from which SnapManager is executed.

-port *db_port*

Specifies the TCP port number used to access the database that the profile describes.

-osaccount *account_name*

Specifies the name of the Oracle database user account. SnapManager uses this account 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`.

-osgroup *group_name*

Specifies the name of the Oracle database group name associated with the oracle account.

-rman

The options that follow `-rman` specify the details that SnapManager uses to catalog backups with RMAN.

-tnsname *tnsname*

Specifies the tnsname connection name (this is defined in the `tnsname.ora` file).

-login

Starts the RMAN login details.

-password *rman_password*

Specifies the password used to log in to the RMAN catalog.

-username *rman_username*

Specifies the user name used to log in to the RMAN catalog.

-controlfile

Specifies that SnapManager uses the control files instead of the RMAN catalog.

-asm

Specifies that SnapManager uses the Oracle Automatic Storage Management (ASM) instance. The options that follow `-asm` specify the details that SnapManager uses to connect to ASM.

-dbname *asm_dbname*

Specifies the name of the ASM database instance.

-host *asm_host*

Specifies the name or IP address of the host the ASM database instance is running on.

-login

Starts the ASM login details.

-password *asm_password*

Specifies the password used to connect to ASM.

-username *asm_username*

Specifies the user name used to connect to ASM.

-port *asm_port*

Specifies the TCP port number used to connect to ASM.

-sid *asm_sid*

Specifies the SID of the ASM database. 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 -sid option. For example, if you are using RAC, you must specify the SID of the RAC instance on the RAC node from which SnapManager is executed.

-retain *number*

Specifies the number of successful backups to retain before automatically deleting the oldest backups. If you do not specify a value, SnapManager replaces the previous backup each time it creates a new backup.

This option counts backups that were not used to clone a database and were successful. SnapManager does not delete backups used to create clones or unsuccessful backups. Refer to “[SnapManager’s policy for retaining backups](#)” on page 21 for more information about the retention policy.

Note

Although SnapManager does not limit the -retain value, if you keep too many backups you can reach the snapshot limit of the storage system or run out of disk space.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo profile create command

The following example command creates a profile named SALES1.

```
smo profile create -profile SALES1  
-profile-password sales1  
-repository -dbname SNPMREPO  
-login -username admin1  
-port 1521 -host server1  
-database -dbname SALESDB  
-login -username admin2 -password oracle  
-osaccount oracle -osgroup db3  
-port 1521 -host server2  
-retain 5  
-verbose
```

Output:

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

The smo profile delete command

Using the smo profile delete command

The `smo profile delete` command deletes a profile of a database. Before deleting a profile, you must delete all backups and clones that use the profile.

Syntax:

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

Options:

-profile *profile_name*

Specifies the profile you want to delete.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo profile delete command

The following example deletes the profile named SALES1.

```
smo profile delete -profile SALES1
```

Output:

```
Operation Id [Ncaf00af0242b3e8dba5c68a57a5ae932] succeeded.
```

The smo profile dump command

Using the smo profile dump command

The `smo profile dump` command creates two dump files containing information about backups. There is one file for the client and one for the server. The command returns the paths to the dump files. These files can be helpful if you need to troubleshoot a problem with the operation.

Syntax:

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

Options:

-profile *profile_name*

Specifies the profile for which you want to create the dump files. The profile contains the identifier of the database and other database information.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo profile dump command

The following example creates a dump for the profile `SALES1`.

```
smo profile dump -profile SALES1
```

Output:

```
SMO-15035: Client dump file is located at
"server.domain.com:/u/admin/.netapp/smo/2.1.0/admin_smo_client_dum
p_results.jar".
Server dump file is located at
"server.domain.com:/tmp/smo_dump_results.jar".
```

The smo profile list command

Using the smo profile list command

The `smo profile list` command displays a list of the current profiles. Use the `-verbose` option to display detailed information about each profile.

Syntax:

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

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Displays details about each profile. Also specifies that error, warning, and informational messages are displayed on the console.

Example smo profile list command

The following example displays existing profiles with their details.

```
smo profile list -verbose
```

Output:

```
Profile name: SALES1  
  Database name: SALESDB  
  SID: SALESDB  
  Host: server1.domain.com  
  Port: 1,521  
  Username: admin1  
  Password: *****  
Profile name: SHIP1  
  Database name: SHIPDB  
  SID: SHIPDB  
  Host: server2.domain.com  
  Port: 1,521  
  Username: admin1  
  Password: *****
```

The smo profile show command

Using the smo profile show command

The `smo profile show` command displays information about a profile.

Syntax:

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

Options:

-profile *profile_name*

Specifies the profile to display. The profile contains the identifier of the database and other database information.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo profile show command

The following example shows the details of the `SALES1` profile.

```
smo profile show -profile SALES1
```

Output:

```
Profile name: SALES1
Target database:
  Database name: SALESDB
  SID: SALESDB
  Host: server1
  Port: 1,521
  Username: sys
  Password: *****
Repository:
  Database name: SMOREPO
  SID: SMOREPO
  Host: server1.domain.com
  Port: 1,521
```

```
Username: admin1
Password: *****
RMAN:
  Profile is not registering backups with RMAN
ASM:
  Target database is not using ASM
Oracle user account: oracle
Oracle user group: db3
Backups to retain: 5
```

The smo profile sync command

Using the smo profile sync command

The `smo profile sync` command loads the profile to repository mappings for that repository to a file in your home directory on the local host.

Syntax:

```
smo profile sync
  -repository -dbname dbname
  -host host
  -port port
  -login -username username
  [-quiet | -verbose]
```

Options:

-repository

-dbname *db_name*

Specifies the repository database for the profile to synchronize.

-host

Specifies the database host.

-port

Specifies the port for the host.

-login

Specifies the log in process for the host user.

-username

Specifies the username for the host.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo profile sync command

The following example shows the result of the command to synchronize the profile to repository mappings for the database.

```
smo profile sync -repository -dbname smrepo -host bismarck -port  
1521 -login -username user2
```

Output:

```
SMO-12345 [INFO ]: Loading profile mappings for repository  
"user2@bismarck:smrepo" into cache for OS User "admin".  
Operation Id [Nff8080810da9018f010da901a0170001] succeeded.
```

Now you can enter commands against profiles in the synchronized repository and they are routed to the appropriate host.

The smo profile update command

Using the smo profile update command

The `smo profile update` command modifies the information in an existing profile. The profile must already exist.

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.

Syntax:

```
smo profile update
  -profile profile_name [-profile-password profile_password]
  -database
    -dbname db_dbname -host db_host
    -login -username db_username -password db_password
    [-sid db_sid] -port db_port
  -osaccount account_name
  -osgroup group
  [-rman
    {-tnsname tnsname
    -login -username rman_username -password rman_password} |
    -controlfile]
  [-asm
    -dbname asm_dbname
    -host asm_host
    -login -username asm_username -password asm_password
    -port asm_port -sid asm_sid]
  [-retain number]
  [-remove-asm]
  [-remove-rman]
  [-quiet | -verbose]
```

Options:

-profile *profile_name*

Specifies the name of the profile. This name can be up to 30 characters long and must be unique within the host.

-profile-password *profile_password*

Specify the password for the profile,

-database

The options that follow **-database** specify the details of the database that the profile describes. This is the database that will be backed up, restored, and so on.

-dbname *db_dbname*

Specifies the name of the database that the profile describes. Use either the global name or the SID.

-host *db_host*

Specifies the name or IP address of the host computer the database runs on.

-login

Starts the database login details.

-username *db_username*

Specifies the user name needed to access the database that the profile describes.

-password *db_password*

Specifies the password needed to access the database that the profile describes.

-sid *db_sid*

Specifies the SID of the database that the profile describes. 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 **-sid** option. For example, if you are using Oracle RAC, you must specify the SID of the RAC instance on the RAC node from which SnapManager is executed.

-port *db_port*

Specifies the TCP port number used to access the database that the profile describes.

-osaccount *account_name*

Specifies the name of the Oracle database user account. SnapManager uses this account 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`.

-osgroup *group*

Specifies the group associated with the Oracle database user account.

-rman

The options that follow **-rman** specify the details that SnapManager uses to catalog backups with RMAN.

-tnsname *tnsname*

Specifies the tnsname connection name (this is defined in the `tnsname.ora` file).

-login

Starts the RMAN login details.

-password *rman_password*

Specifies the password used to log in to the RMAN catalog.

-username *rman_username*

Specifies the user name used to log in to the RMAN catalog.

-controlfile

Specifies that SnapManager uses the control files instead of the RMAN catalog.

-asm

.Specifies that SnapManager uses the Oracle Automatic Storage Management (ASM) instance. The options that follow **-asm** specify the details that SnapManager uses to connect to ASM.

-dbname *asm_dbname*

Specifies the name of the ASM database instance.

-host *asm_host*

Specifies the name or IP address of the host the ASM database instance is running on.

-login

Starts the ASM login details.

-password *asm_password*

Specifies the password used to connect to ASM.

-username *asm_username*

Specifies the user name used to connect to ASM.

-port *asm_port*

Specifies the TCP port number used to connect to ASM.

-sid *asm_sid*

Specifies the SID of the ASM database. 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 **-sid** option. For example, if you are using RAC,

you must specify the SID of the RAC instance on the RAC node from which SnapManager is executed.

-retain *number*

Specifies the number of successful backups to retain before automatically deleting backups. If you do not specify a value, SnapManager replaces the previous backup each time it creates a new backup. Only successful backups that are not mounted or used to create a clone are counted in this number. Refer to [“SnapManager’s policy for retaining backups”](#) on page 21 for more information.

Note

Although SnapManager does not limit the -retain value, if you keep too many backups you can reach the snapshot limit of the storage system or run out of disk space.

-remove-asm

Removes the ASM credentials from the profile.

-remove-rman

Removes the RMAN credentials from the profile.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

**Example smo
profile update
command**

The following example changes the login information for the database described by the SALES1 profile. You would run this command *after* changing the login information on the database itself.

```
smo profile update -profile SALES1 -database -dbname SALESDB -sid  
SALESDB -login -username admin2 -password d4jPe7bw -port 1521  
-host server1
```

Output:

```
Operation Id [8abc01ec0e78ec33010e78ec3b410001] succeeded.
```

The smo profile verify command

Using the smo profile verify command

The `smo profile verify` command confirms the profile set up.

Syntax:

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

Options:

-profile *profile_name*

Specifies the profile to verify. The profile contains the identifier of the database and other database information.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo profile verify command

The following example verifies the SALES1 profile.

```
smo profile verify -profile SALES1
```

Output:

```
Operation Id [Nd7169f567e8a4c6c9700b1ebac49993b] succeeded.
```

The smo repository create command

Using the smo repository create command

The `smo repository create` command creates a repository in which to store database profiles and associated credentials. The repository must be stored on an Oracle database instance that is separate from the databases that are backed up and restored using SnapManager.

Syntax:

```
smo repository create
  -repository
    -dbname repo_dbname -host repo_host
    [-login -username username
    -port repo_port]
  [-force] [-noprompt]
  [-quiet | -verbose]
```

Options:

-repository

The options that follow `-repository` specify the details of the database for the repository:

-dbname *repo_dbname*

Specifies the name of the database that stores the repository. Use either the global name or the SID.

-host *repo_host*

Specifies the name or IP address of the host computer the repository database runs on.

-login

Starts the repository login details. The `-login -username db_username -port db_port` are optional. If not specified, SnapManager defaults to OS Authentication Connection Mode.

-username *repo_username*

Specifies the user name needed to access the database that stores the repository.

-port *repo_port*

Specifies the TCP port number used to access the database that stores the repository.

-force

Specifies that SnapManager attempts to force the creation of the repository. Using this option results in SnapManager prompting you to backup the repository before creating the repository.

-noprompt

Specifies that SnapManager not issue the prompt to backup the repository before creating it if you use the `-force` option. Using the `-noprompt` option insures the prompt does not appear, making it easier to create repositories using a script.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo repository create command

The following example creates a repository in the database SALESDB on the host server1.

```
smo repository create -repository -dbname SALESDB -host server1 -  
login -username admin -port 1521  
-force -noprompt
```

The smo repository delete command

Using the smo repository delete command

The `smo repository delete` command deletes a repository used to store database profiles and associated credentials.

Syntax:

```
smo repository delete
  -repository
    -dbname repo_dbname -host repo_host
    -login -username username
    -port repo_port
  [-force] [-noprompt]
  [-quiet | -verbose]
```

Options:

-repository

The options that follow `-repository` specify the details of the database for the repository:

-dbname *repo_dbname*

Specifies the name of the database that stores the repository. Use either the global name or the SID.

-host *repo_host*

Specifies the name or IP address of the host computer the repository database runs on.

-login

Starts the repository login details.

-username *repo_username*

Specifies the user name needed to access the database that stores the repository.

-port *repo_port*

Specifies the TCP port number used to access the database that stores the repository.

-force

Specifies that SnapManager attempts to force the deletion of the repository, even if there are incomplete operations. SnapManager issues a prompt if there are incomplete operations, asking if you are sure you want to delete the repository.

-noprompt

Specifies that SnapManager not prompt you before deleting the repository. Using the `-noprompt` option insures the prompt does not appear, making it easier to delete repositories using a script.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo repository delete command

The following example deletes the repository in the database named SALESDB.

```
smo repository delete -repository -dbname SALESDB  
-host server1 -login -username admin -port 1521
```

The smo repository show command

Using the smo repository show command

The `smo repository show` command displays information about the repository.

Syntax:

```
smo repository show
  -repository
    -dbname repo_dbname -host repo_host -port port
    -login -username username
  [-quiet | -verbose]
```

Options:

-repository

The options that follow `-repository` specify the details of the database for the repository:

-dbname *repo_dbname*

Specifies the name of the database that stores the repository. Use either the global name or the SID.

-host *repo_host*

Specifies the name or IP address of the host computer the repository database runs on.

-port *repo_port*

Specifies the TCP port number used to access the database that stores the repository.

-login

Starts the repository login details.

-username *repo_username*

Specifies the user name needed to access the database that stores the repository.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo repository show command

The following example shows details about the repository in the SALESDB database.

```
smo repository show -repository -dbname SALESDB  
-host server1 -port 1521 -login -username admin
```

Output:

```
Repository Definition:  
  User Name: admin  
  Host Name: server1  
  Database Name: SALESDB  
  Database Port: 1521  
  Version: 28  
Hosts that have run operations using this repository: 2  
  server2  
  server3  
Profiles defined in this repository: 2  
  GSF5A  
  GSF3A  
Incomplete Operations: 0
```

The smo repository update command

Using the smo repository update command

The `smo repository update` command updates the repository that stores database profiles and associated credentials when you upgrade SnapManager.

Attention

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

Syntax:

```
smo repository update
  -repository
    -dbname repo_dbname -host repo_host
    -login -username username
    -port repo_port
  [-force] [-noprompt]
  [-quiet | -verbose]
```

Options:

-repository

The options that follow `-repository` specify the details of the database for the repository:

-dbname *repo_dbname*

Specifies the name of the database that stores the repository. Use either the global name or the SID.

-host *repo_host*

Specifies the name or IP address of the host computer the repository database runs on.

-login

Starts the repository login details.

-username *repo_username*

Specifies the user name needed to access the database that stores the repository.

-port *repo_port*

Specifies the TCP port number used to access the database that stores the repository.

-force

Specifies that SnapManager attempts to force the update of the repository. SnapManager prompts you to make a backup of the current repository before updating.

-noprompt

Specifies that SnapManager not issue the prompt before updating the repository database. Using the `-noprompt` option insures the prompt does not appear, making it easier to update repositories using a script.

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

**Example smo
repository update
command**

The following example updates the repository in the SALESDB database.

```
smo repository update -repository -dbname SALESDB  
-host server1 -login -username admin -port 1521
```

The smo_server restart command

Using the smo_server restart command

The `smo_server restart` command restarts the SnapManager for Oracle host server. Enter this command as root.

Syntax:

```
smo_server restart  
    [-quiet | -verbose]
```

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo_server restart command

The following example restarts the host server.

```
smo_server restart
```

The smo_server start command

Using the smo_server start command

The `smo_server start` command starts the host server running the SnapManager for Oracle software. Enter this command as root.

Syntax:

```
smo_server start  
    [-quiet | -verbose]
```

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo_server start command

The following example starts the host server.

```
smo_server start
```

```
SMO-17100: SnapManager Server started on secure port 25204 with PID  
11250
```

The smo_server status command

Using the smo_server status command

The `smo_server status` command verifies the status of the SnapManager for Oracle host server. Enter this command as root.

Syntax:

```
smo_server status  
    [-quiet | -verbose]
```

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo_server status command

The following example requests the status of the host server.

```
smo_server status
```

```
SMO-17104: SnapManager Server version 2.1.0 is running on secure  
port 25204 with PID 11250 and has 0 operations in progress.
```

The smo_server stop command

Using the smo_server stop command

The `smo_server stop` command stops the SnapManager for Oracle host server. Enter this command as root.

Syntax:

```
smo_server stop  
    [-quiet | -verbose]
```

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo_server stop command

The following example stops the host server.

```
smo_server stop
```

The smo system dump command

Using the smo system dump command

The `smo system dump` command creates two dump files containing information about the environment. There is one file for the client and one for the server. These files can be helpful if you need to troubleshoot a problem with the operation.

Syntax:

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

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo system dump command

The following example creates system dump files.

```
smo system dump
```

Output::

```
SMO-15035: Client dump file is located at  
"server.domain.com:/u/admin/.netapp/smo/2.1.0/admin_smo_client_dum  
p_results.jar".  
Server dump file is located at  
"server.domain.com:/tmp/smo_dump_results.jar".
```

The smo system verify command

Using the smo system verify command

The `smo system verify` command confirms that all the components of the environment required to run SnapManager are set up correctly.

Syntax:

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

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Specifies that error, warning, and informational messages are displayed on the console.

Example smo system verify command

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

```
smo system verify -verbose
```

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.

Syntax:

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

Options:

-quiet

Specifies that only error messages are displayed on the console. The default is to display error and warning messages.

-verbose

Displays the build date and the contents of each profile. Also specifies that error, warning, and informational messages are displayed on the console.

Example smo version command

The following example displays the installed version of SnapManager.

```
smo version
```

Output:

```
SnapManager for Oracle Version: 2.1
```

Index

A

- abort operation
 - executing from CLI 115
 - general steps 115
 - graceful cleanup 114
 - incomplete backup 114
 - interrupting executing operation 114
- aborting
 - operations 217
- ASM
 - backup fails with raw LUNs on qtree 166
 - confirming it is accessible 74
 - Linux limitation 32
- ASM, RAC, and RMAN integration 4
- auto recovery
 - defined 5

B

- backup 21
 - cloning 92
 - defined 5
 - examples of retaining 21
 - problem seeing in RMAN 165
 - removing 129
 - repository 10
 - retaining
 - retaining backups 21
 - sliding window 21
 - steps to display information 121, 123
 - unable to display in GUI 154
 - verifying 69, 85
- backup create
 - command parameters 180
 - example 183
- backup delete
 - command parameters 184
 - example 185
- backup list
 - command parameters 186
 - example 186
- backup mount
 - command parameters 188
 - example 188
- backup operation
 - algorithm for full backup 18
 - algorithm for partial backup 19
 - executing from CLI 84
 - fails without message 167
 - files included 12
 - full 17
 - hangs 166
 - incomplete 114
 - information required 82
 - offline backup 82
 - online backup 82
 - partial 17
 - types of backups 12
 - viewing files 12
- backup restore
 - command parameters 190
 - example 192
- backup show
 - command parameters 193
 - example 193
- backup unmount
 - command parameters 198
 - example 199
- backup update
 - command parameters 196
 - example 197
- backup verify
 - command parameters 200
 - example 201
- backup verify operation
 - executing from CLI 85
 - steps 85
- backups
 - displaying information about 121
 - removing 129
- before issuing commands
 - start server 64

C

- case-sensitive
 - command options 178
- changing
 - existing profiles 119
 - profile properties 119
- clearing
 - credentials 212
- CLI
 - executing abort operation 115
 - executing backup operation 84
 - executing backup verify operation 85
 - executing clone operation 104, 107
 - executing list operation 121, 124
 - executing mount operation 132
 - executing remove operation 130
 - executing restore operation 88
 - executing unmount operation 132
 - modifying profile 119
 - profile create command 63, 76
 - profile delete command 120
 - starting SnapManager 37
- clone
 - problem with mount point 156
- clone create
 - command parameters 202
 - example 203
- clone delete
 - command parameters 204
 - example 204
- clone list
 - command parameters 205
 - example 205
- clone operation
 - executing from CLI 104, 107
 - fails with NID-00100 156
- clone show
 - command parameters 206
 - example 206
- Clone Specification 95
- clone specification file
 - creating 95
- clone template
 - command parameters 208
 - example 208
- cloning backups 92
 - clone specification file 95
 - cloning RAC database 108
 - creating Oracle files 93
 - different SIDs 93
 - files created 93
 - RAC database 93
 - setting up RAC clone 109
- cmdfile
 - command parameters 210
- cmdfile command
 - example 210
- command options
 - case-sensitive 178
 - displaying messages 179
 - quiet 179
 - verbose 179
- commands
 - backup create 180
 - backup delete 184
 - backup list 186
 - backup mount 188
 - backup restore 190
 - backup show 193
 - backup unmount 198
 - backup update 196
 - backup verify 200
 - clone create 202
 - clone delete 204
 - clone list 205
 - clone show 206
 - clone template 208
 - cmdfile 210
 - credential clear 212
 - credential list 213
 - credential set 214
 - general information 177
 - help 216
 - operation abort 217
 - operation dump 219
 - Oracle dbv 69, 85
 - profile create 221
 - profile delete 226
 - profile dump 227
 - profile list 228

- profile show 229
- profile sync 231
- profile update 233
- profile verify 237
- recover option 190
- repository delete 240
- repository show 242
- repository update 244
- server restart 246
- server start 247
- server status 248
- server stop 249
- system dump 250
- system verify 251
- version 252
- Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products 173
- Components 95
- controlfile
 - recovery slow 165
- controlfiles
 - included in backup 12
- conventions
 - formatting xi
 - keyboard xii
- creating
 - backups 180
 - clone template 208
 - clones 202
 - cmdfile 210
 - operation dump files 219
 - profile dump file 227
 - profiles 73, 221
 - system dump files 250
- credential clear
 - command parameters 212
 - example 212
- credential list
 - command parameters 213
 - example 213
- credential set
 - command parameters 214
 - example 215

D

- Data ONTAP Block Access Management Guide 174
- database
 - confirming it is accessible 74
 - recovering 87
 - restoring 87
- databases
 - cloning 93
 - RAC 93
- DBA
 - using SnapManager 2
- delete
 - profile 120
 - repository 118
- deleting
 - backups 184
 - clones 204
 - profile 120
 - profiles 120, 226
 - repositories 118, 240
- Documentation
 - NOW Description Page 173
- documentation
 - additional 173
 - Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products 173
 - Data ONTAP Block Access Management Guide 174
 - FCP host attach kit 174
 - SnapDrive for UNIX Quick Start 174
 - SnapManager Compatibility Matrix 173
 - SnapManager Release Notes 173
 - System Configuration Guide 174
- dump command
 - executing 151
 - information produced 150
- dump files
 - creating 151
 - locating 150

E

- environment

- verifying 73, 74
- environment variables
 - checking 69
 - verifying
 - checking 74
- error messages 179
 - code ranges 171
- examples
 - backup create command 183
 - backup delete command 185
 - backup list command 186
 - backup mount command 188
 - backup restore command 192
 - backup show command 193
 - backup unmount command 199
 - backup upate command 197
 - backup verify command 201
 - clone create command 203
 - clone delete command 204
 - clone list command 205
 - clone show command 206
 - clone template command 208
 - cmdfile command 210
 - credential clear command 212
 - credential list command 213
 - credential set command 215
 - help command 216
 - operation abort command 218
 - operation dump command 219
 - profile create command 225
 - profile delete command 226
 - profile dump command 227
 - profile list command 228
 - profile show command 229
 - profile sync command 232
 - profile update command 236
 - profile verify command 237
 - recover option 192
 - repository create command 239
 - repository delete command 241
 - repository update command 243, 245
 - retaining backups 21
 - server restart command 246
 - server start command 247
 - server status command 248

- server stop command 249
- system dump command 250
- system verify command 251
- version command 252

F

- file system
 - full 167
- full backup
 - algorithm 18
 - compared to partial 17
 - defined 5

G

- getting
 - command help 216
- GUI
 - installing 52
 - unable to display backups 154
- GUID
 - defined 5
 - view using list 177

H

- help
 - command parameters 216
- help command
 - example 216
- hosts
 - installation steps on Linux host 51

I

- in-band events 24
- informational messages 179
- installing
 - GUI 52
 - steps for Linux host 51

L

- label option
 - case sensitive 177

- licenses
 - required 47
- limitation 32
- Linux host
 - installation steps for SnapManager 51
 - requires ASMLib 32
 - uninstalling SnapManager 139
- list operation
 - executing from CLI 121, 124
 - general steps 121, 123
- listing
 - backups 186
 - clones 205
 - credentials 213
 - profiles 228
- log files
 - file system full 167
 - included in backup 12

M

- management
 - tasks 111
- managing
 - SnapManager operations 111
- mount operation
 - executing from CLI 132
 - general steps 131
- mount point
 - problem with clone 156
- mounting
 - backups 188

N

- NID-00100 error 156
- NOW site
 - downloading software 49

O

- OCFS
 - unsupported 32
- offline
 - defined 5
- offline backup 82

- online
 - defined 5
- online backup 82
- operation abort
 - command parameters 217
 - example 218
- operation dump
 - command parameters 219
 - example 219
- operational states
 - failed 23
 - in progress 23
 - in-band (recoverable) 24
 - incomplete 23
 - out-of-band (unrecoverable) 24
 - SnapManager 23
 - succeeded 23
- operations
 - managing SnapManager 111
- out-of-band
 - events 24
- Overview, ASM, RAC, and RMAN integration 4
- Overview, repositories and profiles 5

P

- partial backup
 - algorithm 19
 - compared to full 17
 - defined 5
- persistent state
 - maintained by SnapManager 114
- PIT recovery
 - defined 5
- profile
 - changing existing 119
 - create 73
 - creating dump files 227
 - creating from CLI 63, 76
 - defined 5
 - delete 120
 - deleting from CLI 120
 - explained 8
 - information needed to create 75
 - modifying from CLI 119

- verifying 73
- profile create
 - command parameters 221
 - example 225
- profile delete
 - command parameters 226
 - example 226
- profile dump
 - command parameters 227
 - example 227
- profile list
 - command parameters 228
 - example 228
- profile show
 - command parameters 229, 231
 - example 229
- profile sync
 - example 232
- profile update
 - command parameters 233
 - example 236
- profile verify
 - command parameters 237
 - example 237
- profiles
 - changing existing 119
- properties
 - profile 119
- protocol
 - license required 47
- protocols
 - accepted 29

Q

- qtrees
 - ASM issue with raw LUNs 166
- quiet command option
 - using 179

R

- RAC database
 - clone options 108
 - cloning 93, 108
 - setting up clone 109

- specify one node 9
- recover
 - defined 5
 - option
 - general steps 88
- recover option
 - auto 5
 - command parameters 190
 - example 192
 - point-in-time 5
 - recovering data 87
 - used with restore 87
 - using with restore command 88
- recoverable events 24
- recovery operations
 - types 12
- remove operation
 - executing from CLI 130
 - general steps 129
- repositories
 - deleting 118
 - multiple 10
- repository
 - backing up 10
 - defined 6
 - delete 118
 - delete repositories
 - deleting 118
 - explained 8
 - information needed to create 60
- repository create
 - example 239
- repository delete
 - command parameters 240
 - example 241
- repository show example 243
- repository update
 - command parameters 244
 - example 245
- respository show
 - command parameters 242
- restarting
 - server 246
- restore
 - defined 6

- restore operation
 - executing from CLI 88
 - general steps 88
 - options 87
 - types 12
- restoring
 - backups 190
- retaining backups
 - examples 21
- RMAN
 - confirming it runs 74
 - problem seeing backups 165

S

- server
 - starting 64
- server restart
 - command parameters 246, 247
 - example 246
- server start
 - example 247
- server status
 - command parameters 248
 - example 248
- server stop
 - command parameters 249
 - example 249
- setting
 - credentials 214
- shell
 - using cmdfile to enter commands 177
- showing
 - backups 193
 - clones 206
 - profiles 229
 - repositories 242
- SIDs
 - used with clones 93
- sliding window 21
- SnapDrive for UNIX
 - required 29
- SnapManager
 - additional documentation 173
 - backing up data 82

- checking the version 138
- downloading software 49
- getting 49
- installation steps on Linux 51
- installing GUI 52
- licenses 47
- limitations 32
- operational states 23
- starting from CLI 37
- uninstalling from Linux host 139
- uninstalling the GUI 139
- working with RAC 9
- SnapManager Compatibility Matrix 173
- SnapManager Release Notes 173
- SnapRestore
 - required 47
- snapshot
 - defined 6
- software
 - downloading 49
- SQLPlus
 - confirming it runs 69, 74
- starting
 - server 247
 - the server 64
- stopping
 - server 249
- storage manager
 - using SnapManager 2
- synchronizing
 - profiles 231
- System Configuration Guide 174
- system dump
 - command parameters 250
 - example 250
- system verify
 - command parameters 251
 - example 251

T

- tasks
 - managing SnapManager 111
- terms
 - auto recovery 5

- backup 5
- full backup 5
- GUID 5
- offline 5
- online 5
- partial backup 5
- PIT recovery 5
- profile 5
- recover 5
- repository 6
- restore 6
- snapshot 6
- troubleshooting
 - ASM backup fails 166
 - backup fails without message 167
 - backup hangs 166
 - clone mount point issue 156
 - controlfile recovery slow 165
 - file system 167
 - NID-00100 error 156
 - problem seeing RMAN backups 165
 - unable to display GUI backups 154
 - using the dump command 150

U

- Understanding repositories and profiles 5
- uninstalling
 - SnapManager from Linux host 139
 - SnapManager GUI 139
- unmount operation
 - executing from CLI 132
 - general steps 131

- unmounting
 - backups 198
- unrecoverable events 24
- updating
 - backups 196
 - profiles 233
 - repositories 244

V

- verbose command option
 - using 179
- verify
 - backup 85
 - environment 73
- verifying
 - backups 69, 200
 - profile 73
 - profiles 237
 - server status 248
 - system 251
 - version information 252
- version
 - checking 138
 - command parameters 252
 - example 252

W

- warning messages 179
- Windows system
 - installing GUI 52
 - uninstalling SnapManager GUI 139