

SnapManager® 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 215-01709_B0
June 2006

Copyright and trademark information

Copyright information

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

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

Portions of this product are derived from the Berkeley BLOG Inference Engine, which is copyrighted and publicly distributed by The Regents of the University of California. As a product of the Berkeley Computer Science Division, the BLOG Inference Engine is released under the BSD license:

Copyright (c) 2005, 2006, Regents of the University of California. All rights reserved.

Software derived from copyrighted material of The Regents of the University of California is subject to the following license and disclaimer:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the University of California, Berkeley nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "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 THE COPYRIGHT OWNER OR CONTRIBUTORS 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.

The BLOG Inference Engine also includes several pieces of third-party software: the CUP parser generator (version 10k, not the new Java 1.5 version); the JLex lexical analyzer generator; and the JAMA matrix package. These third-party packages have their own open-source licenses.

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

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

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

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

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

Network Appliance NetCache is certified RealSystem compatible.

Table of Contents

	Preface	ix
Chapter 1	Overview of SnapManager for Oracle	1
	Before you begin: A quick look at some SnapManager basics.	2
	Using SnapManager	7
	Where to go for more information	13
Chapter 2	SnapManager Concepts	15
	Navigation containers used by SnapManager	16
	Working with repositories	19
	Approaches to backing up, restoring, and recovering data	21
	How SnapManager handles partial and complete online backups	26
	SnapManager's policy for retaining backups.	30
	SnapManager operational states	32
	Key points for DBAs and storage administrators.	34
Chapter 3	Installing SnapManager	35
	Installing SnapManager.	36
	Getting and installing the SnapManager software	40
	Installing the SnapManager files on a Linux host	42
	Installing the SnapManager files on a Solaris host.	45
	Add the repository schema and restart the NetApp Host Agent	49
	Install the GUI on a Windows system	52
	Upgrading to a new version of SnapManager	53
Chapter 4	Getting Started with SnapManager	61
	Getting started with SnapManager	62
	Setting up and managing database profiles.	67

	Verify that the system environment is set up correctly.	79
Chapter 5	Backing Up and Restoring Databases	81
	Creating database backups	82
	Verifying a backup	88
	Restoring and recovering a database	90
Chapter 6	Managing SnapManager Operations	97
	Aborting an operation.	98
	Displaying information about backups	101
	Removing backups	104
	Mounting and unmounting backups	107
	Checking the version of SnapManager	111
	Managing log files	112
Chapter 7	Using SnapManager to Clone Databases	113
	General information about cloning backups using SnapManager	114
	Creating a clone map file	116
	Cloning a database	120
	Using cloning with RAC databases.	125
Chapter 8	SnapManager Command Reference	127
	Collecting information needed by SnapManager Commands	128
	Conventions for entering names and command options	129
	Common options for handling messages shown on the console	130
	The smo backup command	131
	The smo check command.	136
	The smo clone command	137
	The smo dump command.	139
	The smo help command	141

The smo manage abort command.	142
The smo manage list command.	144
The smo manage mount command	145
The smo manage remove backup command	147
The smo manage unmount command.	150
The smo pfile command	152
The smo profile create command.	154
The smo profile delete command.	159
The smo profile list command	161
The smo profile show command	162
The smo profile update command	164
The smo restore and recovery command	169
The smo verify command.	173
The smo version command	175
 Chapter 9	
Troubleshooting	177
Questions to ask when you encounter a problem.	178
Using the dump command to gather information.	179
General troubleshooting tips for using the GUI	182
General troubleshooting tips for working with clones	183
General troubleshooting tips for SnapManager	185
Common error conditions.	188
 Index	191

Preface

About this guide	This document describes how to install, configure, and use SnapManage® for Oracle® 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, NetApp Host Agent, Open Systems SnapVault™, or SnapDrive™ for UNIX® documentation.
Audience	This guide is for database administrators who are familiar with Oracle and storage administrators who possess a working knowledge of NetApp® storage systems.
Command conventions	SnapManager for Oracle (SnapManager) lets you enter commands from the Command Line Interface (CLI) on your UNIX host where the target database is running. You can also execute SnapManager from a Graphical User Interface (GUI) running on a Windows host.
Formatting conventions	The following table lists different character formats used in this guide to set off special information.

Formatting convention	Type of information
<i>Italic type</i>	◆ Words or characters that require special attention. ◆ Placeholders for information you must supply. For example, if the guide requires you to enter the <code>fcctest adaptername</code> command, you enter the characters “fcctest” followed by the actual name of the adapter. ◆ Book titles in cross-references.

Formatting convention	Type of information
Monospaced font	◆ Command and daemon names. ◆ Information displayed on the system console or other computer monitors. ◆ The contents of files.
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:

Note

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

Caution

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

WARNING

A warning contains instructions that you must follow to avoid personal injury.

About this chapter This chapter provides information about how SnapManager® for Oracle® (SnapManager) works and the operations it can perform.

Topics in this chapter This chapter includes the following topics:

- ◆ “[Before you begin: A quick look at some SnapManager basics](#)” on page 2
- ◆ “[Using SnapManager](#)” on page 7
- ◆ “[Where to go for more information](#)” on page 13

Before you begin: A quick look at some SnapManager basics

SnapManager basics

To make this document and the SnapManager for Oracle product easier to understand, you should know the following:

- ◆ SnapManager works with backups created using snapshots. This release does not provide disk-to-disk backups. (For information on backups, see “[Key SnapManager terms](#)” on page 4.)
- ◆ SnapManager is part of the NetApp product set. As a result, you can use SnapManager as a stand-alone product, or you can also expand its usefulness by using it with other NetApp products, such as SnapVault. SnapManager takes advantage of existing NetApp tools and technologies, including snapshots.
- ◆ 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.

Note

You must manually configure SnapDrive for UNIX after you install SnapManager and any time you add a new storage system to ensure that it can access the required storage systems.

- ◆ You must have the FlexClone™ license if you want to take full advantage SnapManager with NFS databases. SnapManager uses FlexClone to:
 - ❖ Clone NFS databases.
 - ❖ Register backups of NFS databases with RMAN (if you are using RMAN).

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

- ◆ For a complete list of SnapManager requirements, see the SnapManager & SnapDrive Compatibility Matrix. This matrix is online on the NOW site at <http://now.netapp.com/NOW/knowledge/docs/docs.shtml> in the Interoperability section.

Using this document

This document describes how to set up and use SnapManager. It contains general information about SnapManager as well as specific information on how to install SnapManager, what tasks you can perform, and reference information on the CLI commands.

The following is a breakdown of the types of information in this document:

- ◆ Conceptual. These chapters provide general information about SnapManager and how it works:
 - ❖ Overview of SnapManager for Oracle (the current chapter)
 - ❖ [“SnapManager Concepts”](#) on page 15.
- ◆ Installation and setup information. These chapters provide information on how to install SnapManager and the setup work you need to do before you can start using it, such as creating a profile:
 - ❖ [“Installing SnapManager”](#) on page 35
 - ❖ [“Getting Started with SnapManager”](#) on page 61
- ◆ Tasks. The following chapters contain information on the different SnapManager tasks you can perform:
 - ❖ [“Backing Up and Restoring Databases”](#) on page 81
 - ❖ [“Managing SnapManager Operations”](#) on page 97
 - ❖ [“Using SnapManager to Clone Databases”](#) on page 113
- ◆ Reference. The following chapters contain reference information about the SnapManager commands and information for troubleshooting any issues you might encounter while using SnapManager:
 - ❖ [“SnapManager Command Reference”](#) on page 127
 - ❖ [“Troubleshooting”](#) on page 177

Note

For information on known problems and workarounds for them, see the SnapManager Release Notes.

Performing SnapManager operations

You can execute SnapManager from both the CLI and the GUI. In general, the CLI commands and the GUI screens and wizards use the same information.

To make SnapManager operations easier to perform regardless of whether you are using the CLI or the GUI, this document presents information this way:

1. It organizes the information based on the task you want to perform.

2. Each task section provides a general list of the information required by that operation. Experienced users can use this list to gather the information they need to perform the task regardless of whether they are using the CLI or the GUI.
3. In each task section, there is a subsection containing the CLI command line used to execute that operation and examples of using this command line. This subsection does not provide the details of all the options in the command line. That information is in the general list of information. In addition, a detailed explanation of all the command options is in the chapter “[SnapManager Command Reference](#)” on page 127.
4. Each task section also provides a subsection with step-by-step information on using the GUI to perform the operation. To execute the GUI, you need most of the same general information that the CLI command requires.

Key SnapManager terms

The following terms are important to understanding and using SnapManager.

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

Note

For this release, SnapManager provides backups that are virtual copies of the database instance and are stored on the same physical medium as the database. SnapManager’s backups consist of the snapshots taken at the point in time when the database instance was backed up.

When you create a backup, SnapManager allows you to specify one of the following:

- ◆ Full backup. SnapManager makes a copy of the entire database instance.
- ◆ Partial backup. SnapManager makes a copy of either the tablespaces or files you specify. You cannot include both tablespaces and files in the same partial backup.

GUID: A Globally Unique Identifier (GUID) for each operation that SnapManager automatically generates for each backup. This is a 32-character HEX string.

Offline: The database is not available to users (i.e., it is in cold mode). When you perform certain operations such as backup, SnapManager lets you specify the state of the database at the time the backup is performed. If you use the `-force` option when you execute a backup operation, SnapManager changes the state of the database as necessary. For example, based on the options you supply

for this operation, it might change the state of the database from online to offline. If you want to create a clone of a backup, you must create a full backup while the database is offline.

Online: The database is available to users (i.e., open). When you perform certain operations such as backups, SnapManager lets you specify the state of the database at the time the backup is performed. If you use the `-force` option when you execute a backup operation, SnapManager changes the state of the database as is necessary to perform the backup. For example, based on the options you supply for this operation, it might change the state of the database from offline to online.

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

Recover: An operation that enables you to use a restored database instance. The recover operation is part of the SnapManager restore command. You can perform a SnapManager restore operation and use a different tool to recover the database, or you can perform a SnapManager restore/recover operation. When you include the recover option, SnapManager recovers the database to a known state from a given backup. You can specify that the recovery be based on one of the following:

- ◆ Point-in-time (PIT) recovery. SnapManager recovers the database to the point in time you specify. This can be either an Oracle System Change Number (SCN) or a date and time.
- ◆ Auto recovery. SnapManager recovers the database to the last transaction that occurred in the database (i.e., the last commit).

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

Restore: An operation that replaces the current data (data files, control files, and so on) with the backed up data. When you restore a backup, SnapManager allows you to specify one of the following:

- ◆ Complete restore. SnapManager restores the entire backup.
- ◆ Selective restore. SnapManager restores only the data you specify. You can specify one of the following for each restore operation:
 - ❖ Tablespaces

- ❖ Data files
- ❖ Control files

Snapshot: A point-in-time, read-only copy of the data you wanted backed up. You can use this snapshot to restore the data at a later date if necessary. By default, SnapManager keeps these snapshots of the backup on the same storage system. You can move the snapshots to a different system. If you need to use them for a restore and recovery, you must return them to the original system.

Using SnapManager

What SnapManager does

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

If you prefer to use a Graphical User Interface (GUI), SnapManager provides a GUI you can run on a Windows host. The GUI provides screens and wizards you can use to perform essentially the same operations that you can perform using the CLI. In addition, the GUI lets you manage your SnapManager operations from a remote location.

SnapManager works with NetApp storage systems to enable you to perform the following tasks:

- ◆ Use snapshots to create a backup of data on a NetApp storage system. 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.

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

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

Note

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

- ◆ Restore and recover a full database or a portion of the database. SnapManager lets you restore the full database instance, specific tablespaces, specific files, or control files. Once you have restored the information, you can 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 a SnapManager for both operations.

You can specify the level to which SnapManager restores and recovers information based on what you specify when performing the restoration. For example, you can identify the elements of the database instance you want to recover based on a point in time. This can be either a date and time or an Oracle SCN.

the date and time you enter as an argument or the Oracle SCN for the database.

- ◆ Track details and produce reports.

SnapManager reduces the level of detail you must track. When you specify the database you want 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 lets you display information about the backup, the profile, and the repository.

- ◆ Verify the validity of the backup.

You can use SnapManager to confirm that the backup completed using standard Oracle backup verification operations. You can either perform the verification as part of the backup operation or at a later time.

In addition, SnapManager is seamlessly integrated with NetApp's SnapValidator™.

- ◆ Clone a backup of a database.

SnapManager can quickly create a writable, space-efficient clone of a database instance it has backed up. The clone is separate from the actual backup, so you can modify it without changing the backup source.

SnapManager uses NetApp's FlexClone feature, which is part of Data ONTAP, to create the clone. FlexClone enables SnapManager to use snapshots of the database instance to avoid having to create an entire physical, disk-to-disk copy. The snapshots require less time to create and take up significantly less space than a physical copy.

In general, you can perform operations when the database is online or offline. However, the database must be offline in order to create a clone.

NetApp products that SnapManager uses

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

The products SnapManager works with include:

- ◆ NetApp's SnapDrive for UNIX. It uses this product to create a point-in-time copy of the database. Using snapshots ensures that the backups are space efficient and faster to create than disk-to-disk backups.
- ◆ NetApp Host Agent software.
- ◆ NetApp's FlexClone feature, which is part of Data ONTAP. It uses this feature to create fast, space-efficient clones of backups. You cannot use traditional volumes with SnapManager.

Using SnapManager with Oracle

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

You can set up the profile to interact with ASM and RMAN.

With RAC configurations, you can invoke SnapManager from a RAC database node to create the backup, restore and recover the database, and clone the database.

If you use RMAN, you can have SnapManager catalog the backups it makes with RMAN. Since the backups are registered with RMAN, the database administrator can make use of SnapManager's backups and preserve the value of all RMAN functions, such as block level restore.

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

How SnapManager works

SnapDrive organizes information into profiles and repositories. The profiles hold the information about the database, including its credentials, any backups, and any clones you've made of it. As a result, you only need to supply a profile name when you perform SnapManager operations; you do not need to repeat the credentials required to access the information.

Note

SnapManager stores the credential information in an encrypted file using DES.

The repositories hold the profiles and its backups. The repository must be located in a different database from the one you are backing up. The database holding the repository can be on the same storage system as the database you are backing up

or it can be on a different system. A single repository can hold information on multiple profiles; however, each database is normally associated with only one profile. You can have multiple repositories, where each repository holds multiple profiles.

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

For more information on profiles and repositories, see “[SnapManager Concepts](#)” on page 15.

Methods for executing SnapManager

You can execute SnapManager in either of two ways:

- ◆ On the UNIX host, by issuing commands from the CLI. You can use a single command to perform each operation. For information on SnapManager’s commands and their arguments, see “[SnapManager Command Reference](#)” on page 127.
- ◆ On a Windows host connected with the UNIX host, by executing the GUI. The GUI contains numerous wizards and other screens to step you through different tasks. It also displays information about the operations that have been performed. For more information about using the GUI, see “[Getting started with SnapManager](#)” on page 62.

Prerequisites for using SnapManager

Before you install SnapManager, you must set up your system. You must have done the following before you install SnapManager:

Note

The most up-to-date information about the requirements for your host operating system is on the NOW site. Check out the matrices in the Interoperability section on the <http://now.netapp.com/NOW/knowledge/docs/docs.shtml> page. In particular, review the SnapManager & SnapDrive compatibility matrix, which contains information on the versions you need for each part of the stack.

- ◆ Set up your host system and storage systems. On the host, you must have:
 - ❖ Host operating system and appropriate patches installed. SnapManager works on hosts running either Red Hat® Enterprise Linux® or Solaris®.
 - ❖ Oracle installed. In addition, set up any Oracle components, such as RMAN or ASM, that you are using.
 - ❖ The NetApp Host Agent installed.

- ❖ NetApp's SnapDrive for UNIX installed for your host platform, including the products it requires such as the Network Appliance™ Host Attach Kit (kit) or Host Support Kit (kit). Follow the instructions provided with the kit to set up your storage systems to work with the host.
- ❖ Sun Microsystems® Java Runtime Environment (JRE).

Set up each storage system you're using so that you have:

- ❖ Data ONTAP installed. To use all the functions of SnapManager, you should also have FlexClone licensed.
- ❖ SnapRestore licensed.
- ❖ The correct protocol (FCP, iSCSI, or NFS) installed and licensed. SnapManager runs on a Red Hat Enterprise Linux host using an iSCSI protocol and on a Solaris host using an NFS or FCP protocol.
- ◆ To use the GUI, you must also set up a Windows system. On this system you must have installed:
 - ❖ Sun's JRE.

Keep in mind that SnapManager works with the following:

- ◆ File systems: Raw, ASM, EXT3, VxFS (Veritas®)
- ◆ Volume manager: For Linux®, none; for Solaris, Veritas

Supported configurations

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

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

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

Note

If you need a SnapManager configuration that is not mentioned in the kit documentation, consult with your NetApp sales representative.

**Product limitations,
recommendations**

This release of SnapManager has the following limitations:

- ◆ There is no support for Oracle Cluster File System (OCFS).
- ◆ Linux must use the Automatic Storage Management support library (ASMLib).
- ◆ This version of SnapManager only creates backups using snapshots. The snapshots contain copies of the data that exist on the same storage system as the source data.

Where to go for more information

Relevant documentation

This guide provides information about the basic tasks involved in installing and using SnapManager. If you want 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 “[Getting the software from NetApp](#)” on page 40

- ◆ *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 the SnapManager Description Page. It contains any last-minute information that you need to get your 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.

You can download these documents from the NOW site. See the SAN/IPSAN library at <http://now.netapp.com/NOW/knowledge/docs/san/>.

- ❖ *SnapDrive for UNIX Release Notes*
- ❖ *SnapDrive for UNIX Installation and Administration Guide*
- ❖ *SnapDrive for UNIX Quick Start Guide*

- ◆ The documentation that comes with the NetApp host attach and support kits
You can download these documents from the NOW site. See the SAN/IPSAN library at <http://now.netapp.com/NOW/knowledge/docs/san/>.
- ◆ The documentation that comes with the NetApp Host Agent
You can download these documents from the NOW site. Follow the links from the Download Software page (<http://now.netapp.com/NOW/cgi-bin/software>) for the NetApp Host Agent. The documentation for this product is on its NOW Description Page.
- ◆ *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.comNOW/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 NetApp site (<http://netapp.com/library/tr/>) as well as with vendors who provide NetApp and Oracle products.
- ◆ Host operating system and Oracle database information
NetApp does not provide these documents. See the readme files and other documentation that you received with your host operating system.

About this chapter

This chapter provides information about SnapManager concepts.

This chapter includes the following topics:

- ◆ [“Navigation containers used by SnapManager”](#) on page 16
- ◆ [“Working with repositories”](#) on page 19
- ◆ [“Approaches to backing up, restoring, and recovering data”](#) on page 21
- ◆ [“How SnapManager handles partial and complete online backups”](#) on page 26
- ◆ [“SnapManager’s policy for retaining backups”](#) on page 30
- ◆ [“SnapManager operational states”](#) on page 32
- ◆ [“Key points for DBAs and storage administrators”](#) on page 34

Navigation containers used by SnapManager

Organizing information in SnapManager

SnapManager organizes database and backup information into repositories and profiles. These are the containers that you use when you are backing up and restoring databases.

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

Note

If you were performing this operation from the CLI, you would use the `-profile` option to include the profile name on the command line.

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, what files were backed up, and whether a clone was created from the backup.

SnapManager provides an initial repository schema as part of its installation package. You load this schema into an Oracle database on your system. This can be an existing Oracle database or one that you created for this purpose. For information on setting up your initial repository, see “[Load the schema into the repository database](#)” on page 49.

You can create additional repositories once you have SnapManager running. You can also backup and remove repositories.

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

Understanding profiles

The profile contains the information required for creating the backup and for performing other SnapManager operations as well as information on the operations that have been performed. This information includes the database credentials and how many backups to retain. These credentials include the username/password pairs for accessing the repository, the database, the NetApp Host Agent, and the required connection information for using RMAN or ASM. For more information on what the profile contains, see [“Setting up and managing database profiles”](#) on page 67 or [“The smo profile create command”](#) on page 154.

Note

The credentials are stored in an encrypted file using DES.

SnapManager requires that you create a profile for each database you want to work with. A profile can contain only one database. However, that same database can be defined in more than one profile.

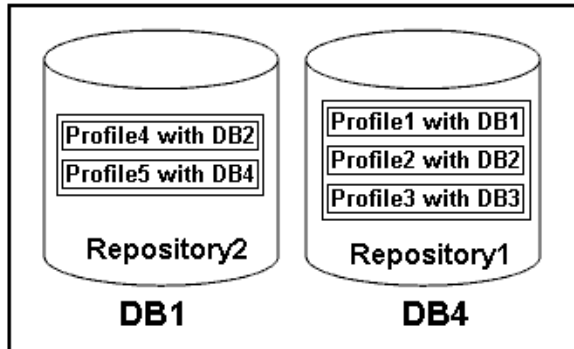
Note

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

The profile also serves as the container for operations such as backups, clones, and so on. You cannot access a backup created using one profile from a different profile, even if both profiles are pointing to the same database.

Once you create the profile, you can easily perform SnapManager operations. You only need to specify the profile name with the SnapManager command; you do not need to re-enter the database credentials each time.

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



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

Working with repositories

Using multiple repositories to make backups of repositories

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

For example, if Repository1 is in Database1, you would have a backup of it each time you created a backup of Database1. Conversely, you would create a backup of Repository2 each time you created a backup of Database2, which contained it. Backing up the repositories this way would protect you 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, suppose you have Repository1 and Repository2. Repository1 is located in the database DB4d, and Repository2 is located in the database DB1a.

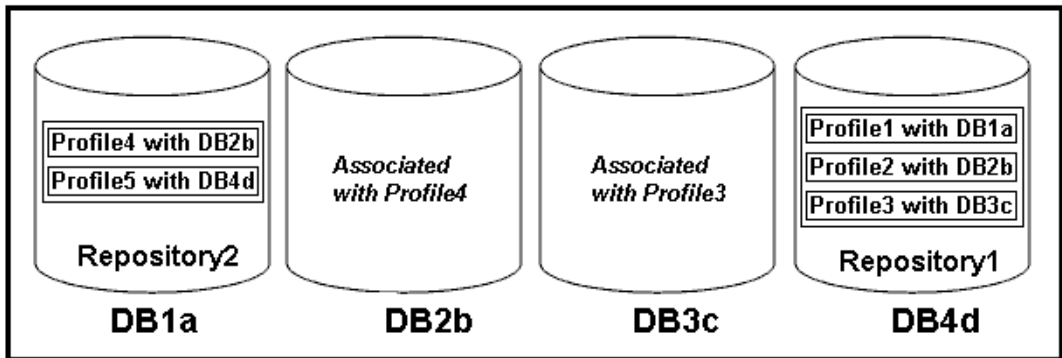
Repository1 has three profiles:

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

Repository2 has two profiles:

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

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



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

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

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

Approaches to backing up, restoring, and recovering data

Types of backup, restore, and recovery operations

SnapManager provides you with several options for backing up, restoring, and recovering data. You can perform these operations using the entire backed up database instance or portions of it. You have the following choices:

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

Note

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

- ◆ Restore operations. You can restore the entire backup or a portion of it. If you perform a selective restore operation, you can 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.

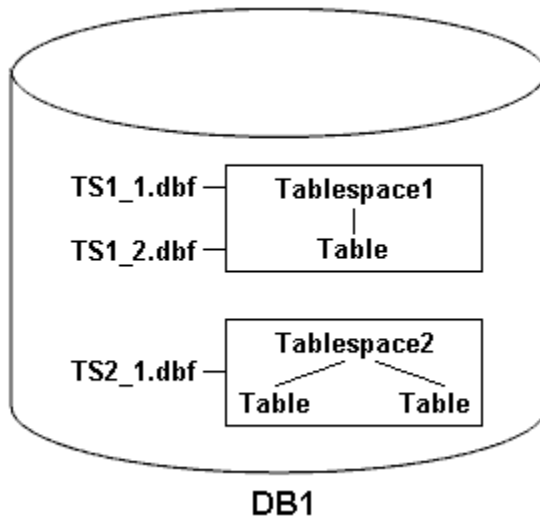
- ◆ Recovery operations. If you decide to use SnapManager to recover the data you restored, you can specify that it recover the data based on either a point in time or auto recovery (i.e., the last transaction committed to the database). The point in time can be an Oracle SCN or a date and time (`yyyy-mm-dd:HH:MM:SS`). By default, SnapManager uses the 24-hour clock.

The following table summarizes the options available with the SnapManager backup, restore, and recover operations. You do not have to use SnapManager to perform the recovery.

Backup	Restore	Recover using SnapManager
Entire database instance Note The database must be in archive log mode if you want to create a backup while the database is running (i.e., online). 	Entire backup	Select either a point-in-time recovery or auto recovery. Point-in-time recovery: ◆ To a specific date ◆ To an SCN Auto recovery: ◆ To the last transaction made on the database instance
Entire database instance	One of the following: ◆ Tablespaces ◆ Data files ◆ Control files	Auto recovery: ◆ 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)	Auto recovery: 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	Auto recovery: To the last transaction made to the database instance

Examples of these operations

Suppose you have a database with two tablespaces. Tablespace1 has one table and two database files associated with it. Tablespace2 has two tables and one database file associated with it.



You have several options for creating backups of this data and then restoring and recovering the data. The following table describes some scenarios you could use.

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.	Point-in-time recovery: SnapManager recovers the data to the point in time you specify: ◆ SCN. You could enter an SCN such as 384641. ◆ Date/time. You could enter a date such as 2005-11-25:19:06:22. Auto recovery: SnapManager recovers the data to the last transaction made to the database. Manual recovery: You perform 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: You specify which information SnapManager restores. ◆ Tablespaces. You can specify both Tablespace1 and Tablespace2 or only one of them. ◆ Data files. You can 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.	Auto recovery: SnapManager recovers the data to the last transaction made to the database. Manual recovery: You perform a manual recovery in Oracle instead of using SnapManager's recover operation.
Partial backup: You specify the data that SnapManager puts in the backup. You can choose one of the following options: ◆ Tablespaces. You can specify both Tablespace1 and Tablespace2 or only one of them. ◆ Data files. You can 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.	Auto recovery: To the last transaction made to the database instance. Manual recovery: You perform a manual recovery in Oracle instead of using SnapManager's recover operation.

Backup	Restore	Recover
Partial backup: You specify the data SnapManager puts in the backup. You can choose one of the following options: ◆ Tablespaces. You can specify both <code>Tablespace1</code> and <code>Tablespace2</code> or only one of them. ◆ Data files. You can specify all three database files (<code>TS1_1.dbf</code>, <code>TS1_2.dbf</code>, and <code>TS2_1.dbf</code>), 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 <code>Tablespace1</code> and <code>Tablespace2</code>, SnapManager would restore only the tablespace you specify. If it contains all three database files (<code>TS1_1.dbf</code>, <code>TS1_2.dbf</code>, and <code>TS2_1.dbf</code>), SnapManager would restore only the data files you specify. If you enter the option <code>-controlfiles</code>, SnapManager only restores the control files.	Auto recovery: To the last transaction made to the database instance. Manual recovery: You perform a manual recovery in Oracle instead of using SnapManager's recover operation.

How SnapManager handles partial and complete online backups

The effect of choosing a full or partial backup

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

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

- ◆ Performing a complete online backup minimizes the number of snapshots. The database can be either online or offline when the backup occurs. 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 backup minimizes the amount of time each tablespace is in backup mode. The database can be either online or offline when the backup occurs. SnapManager groups the snapshots it takes by tablespace. Because each tablespace is in backup mode only long enough to take the snapshots, 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 you set up SnapManager to retain a large number of backups where each backup consists of numerous snapshots.

To compare the differences in how the algorithms create snapshots, suppose the database is two volumes. Each volume has two tablespaces and each tablespace has two database files.

For a full back, SnapManager would take a snapshot of each volume, resulting in a total of two snapshots.

For a partial backup of the same number of database objects, SnapManager would take 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 snapshots produce different numbers 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	

You need to select which type of backup you use based on what is important in your work environment. The trade-offs are the length of time that is acceptable 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 27 and “[Algorithm for performing a partial backup](#)” on page 28

Algorithm for performing a full backup

When you perform a full backup, SnapManager places the entire database in online backup mode. The time that an individual tablespace (for example, /vol/vola/ts1_1.dbf) is in online backup mode is longer than if you specified that only certain tablespaces or data files be backed up.

In addition, because 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 means there is an additional I/O load on the host.

The trade-off is that SnapManager takes fewer snapshots.

When you specify a full backup, SnapManager:

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

As you can see, 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

When you select a partial backup, SnapManager processes the backup one tablespace at a time, even if you specified that it backup data files. SnapManager uses tablespaces because the tablespace level is the lowest level that Oracle allows to be placed 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, you have the trade-off that this process requires more snapshots than performing a full backup. As a result, you risk reaching the limit on the number of snapshots allowed sooner.

For a partial backup, SnapManager takes snapshots of the tablespaces or data files you specify. 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 performs the following tasks:

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

If you perform 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

The following are a few tips that might help you keep an adequate pool of available snapshots so that you reduce the risk of hitting the limit for the maximum number of snapshots.

- ◆ Set up your storage system to maximize the use of LUNs, volumes, and tablespaces. For example, you might try to keep all the LUNs on the same volume.
- ◆ Store the snapshots on a different storage system. If you need to restore backup, you will need to return the snapshots to the original system.
- ◆ Remove backups when you no longer need them. You can set up SnapManager to automatically remove successful backups once it reaches a certain number of them.

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

Note_____

To delete an unsuccessful backup or a cloned backup, you must use SnapManager’s remove operation. You can use either the `smo manage remove` command or the GUI.

When you set the retain option, SnapManager keeps the backups until it has a successful backup that exceeds this number. At that point, it removes the oldest successful backup that was not used to create a clone.

Example of how SnapManager retains backups

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

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

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

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

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

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

On April 14, you take another backup.

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

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

You take two more backups in April that are both successful.

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

SnapManager deletes the oldest successful backups that were not used to create a clone. This keeps the total number of successful backups to three. However, if you were to check the number backups now, you see the following four:

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

SnapManager operational states

Four possible SnapManager operational states

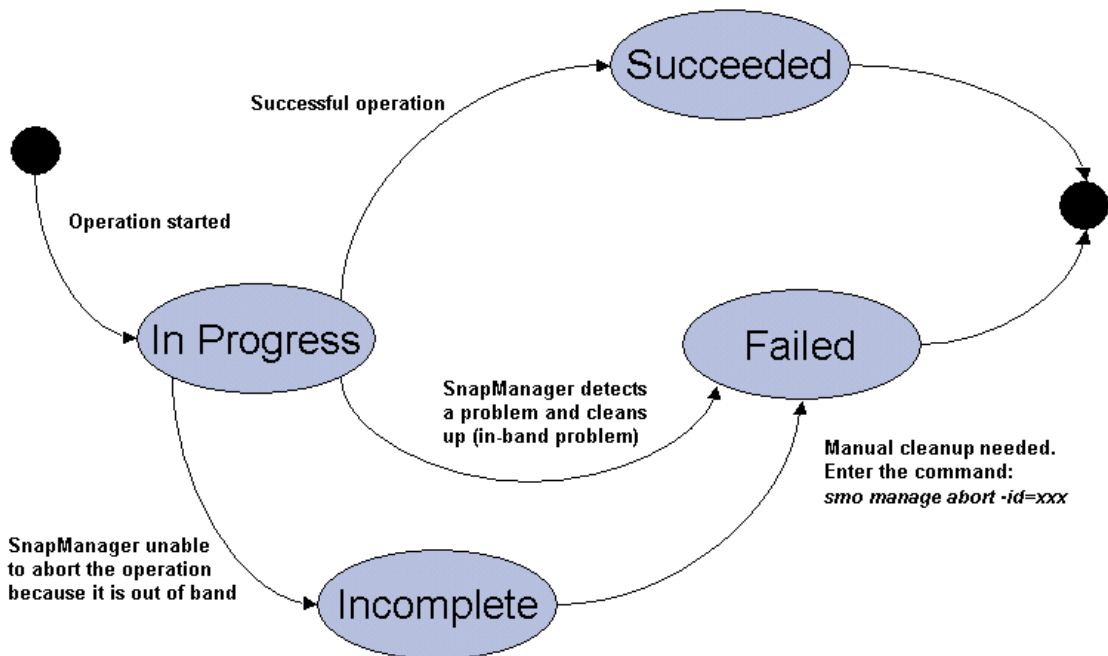
When you execute a SnapManager command, there are four possible operational states that it can enter. They are:

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

`smo manage abort -id id_number`

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

The following illustration shows the operational 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 NetApp storage system.
 - ❖ You are using an ASM database, but the ASM instance is not running.
 - ❖ SnapDrive for UNIX is not installed or cannot access the database.
- ◆ 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.

SnapManager's response to recoverable event

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

SnapManager's response to an out-of-band event

When an out-of-band event occurs, SnapManager cannot execute the abort process, so it kills the process. This leaves SnapManager in an incomplete operational state. To clear this state, you must enter the SnapManager `sмо manage abort` command from the CLI. Enter the process number as the value for the `id` argument. Use the format:

- ◆ `# смо manage abort -id id_number`

Key points for DBAs and storage administrators

SnapManager and administrators

SnapManager's features can benefit both database administrators (DBAs) and storage administrators

Looking at SnapManager from a DBA's perspective

SnapManager is a tool designed to make the lives of DBAs easier by making taking backups easier. SnapManager

- ◆ Speeds the time to create a backup in a space-efficient way, which lets you perform more backups.
- ◆ Organizes information into containers such as profiles to make creating backups and restoring and recovering data quick and easy.
- ◆ Automatically maps the database files to the storage. You do not need to know the underlying storage system.
- ◆ Easily integrates with existing Oracle tools, such as RMAN and ASM.
- ◆ Creates snapshots of logs.
- ◆ Quickly creates a clone from a backup.
- ◆ Reduces the mean time to recover a database by using SnapRestore.

The storage administrator's view of SnapManager

As storage administrator, it is important to know that:

- ◆ SnapManager handles different protocols (FCP, iSCSI, NFS) seamlessly.
- ◆ You can optimize the type of backup (complete or partial) based on what works best in your environment.
- ◆ SnapManager backups are quick and space efficient, so you can do them more frequently, if necessary.
- ◆ Clones are quick and space efficient.
- ◆ SnapManager works seamlessly with host volume managers, such as ASM, Veritas, or LVM.

About this chapter

This chapter provides information about how to download a copy of SnapManager and install it.

Topics in this chapter

This chapter includes the following topics:

- ◆ “[Installing SnapManager](#)” on page 36
- ◆ “[Getting and installing the SnapManager software](#)” on page 40
- ◆ “[Installing the SnapManager files on a Linux host](#)” on page 42
- ◆ “[Installing the SnapManager files on a Solaris host](#)” on page 45
- ◆ “[Add the repository schema and restart the NetApp Host Agent](#)” on page 49
- ◆ “[Install the GUI on a Windows system](#)” on page 52
- ◆ “[Upgrading to a new version of SnapManager](#)” on page 53

Installing SnapManager

Steps required to install SnapManager

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

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

Overview of installing and setting up SnapManager

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

Step	Action	Details
Preparing to install the SnapManager software package		
1	Make sure your system meets the SnapManager requirements. Host prerequisites: ◆ NetApp Host Agent ◆ SnapDrive for UNIX and its requirements ◆ Sun's Java Runtime Environment (JRE) Windows prerequisites: For the Windows system where you want to execute the GUI, you must install: ◆ Sun's JRE Storage system prerequisites: On the storage system, you need to have Data ONTAP installed and running. You should also have the following licenses: ◆ Protocol: FCP, iSCSI, or NFS ◆ FlexClone (if you're using NFS) Note _____ SnapManager uses the FlexClone feature to clone NFS databases and register NFS database backups with RMAN. _____ ◆ SnapRestore	See “Check requirements for SnapManager and its prerequisites” on page 39. There is also information about the documentation available for these NetApp products in the section “Where to go for more information” on page 13.

Step	Action	Details
2	Download a copy of the SnapManager software package as well as a copy of the GUI, if you want to use it. You can also get a copy of the software from the SnapManager CD.	See the section “ Downloading the SnapManager software from NOW ” on page 41.
Installing the SnapManager software package		
3	Follow the instructions for your operating system to install SnapManager. ◆ For Linux, use the <code>rpm</code> command. ◆ For Solaris, use the <code>pkgadd</code> command.	For Linux systems, see the section “Installing the SnapManager files on a Linux host” on page 42. For Solaris platforms, see the section “Installing the SnapManager files on a Solaris host” on page 45. Note _____ You must install SnapManager in its default location.
4	Once you’ve installed the program, run the <code>smo_setup</code> script to set the permissions. Note _____ Linux and Solaris set these permissions by default. Run the <code>smo_setup</code> script if you want to change them.	For Linux systems, see the section “Installing the SnapManager files on a Linux host” on page 42. For Solaris platforms, see the section “Installing the SnapManager files on a Solaris host” on page 45
5	Load the schema into the repository.	See the section “ Add the repository schema and restart the NetApp Host Agent ” on page 49.
6	Restart the NetApp Host Agent.	See the section “ Restarting the NetApp Host Agent ” on page 50.
7	(Optional) Install the SnapManager GUI on a Windows machine.	See the section “ Install the GUI on a Windows system ” on page 52.
Preparing to use SnapManager		
8	Create profiles for the databases you want to back up and restore.	See the section “ Setting up and managing database profiles ” on page 67.

Step	Action	Details
9	Check to make sure all the required components are installed and working with SnapManager.	“ Verify that the system environment is set up correctly ” on page 79.

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, which online in the Interoperability section on the page <http://now.netapp.com/NOW/knowledge/docs/docs.shtml>. (For instructions on getting to this page, follow steps 1 through 4 in the section “[Downloading the SnapManager software from NOW](#)” on page 41.)
- ◆ NetApp Host Agent, go to the NetApp Host Agent NOW page. (For instructions on getting to this page, follow steps 1 through 4 in the section “[Downloading the SnapManager software from NOW](#)” on page 41.)
- ◆ 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>.

Getting and installing the SnapManager software

Getting the software from NetApp

NetApp provides the SnapManager software on both a SnapManager CD and the NOW site. Both the CD and the NOW site contain the software that you install on the UNIX host and the software that you install on the Windows machine where you're running the GUI.

To install the software you need to:

1. Download the host software file for your 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, you must also get a copy of the file containing the GUI software, either by downloading it from the NOW site or from the CD.
4. Place the GUI .zip file on a Windows system and extract the GUI software from it.

Make sure that you have the NetApp Host Agent and SnapDrive for UNIX installed on your UNIX host. If you do not have these products, you can download them from the appropriate locations on the NOW site. Contact your NetApp representative to arrange to purchase these products.

You also need the Sun JRE installed on both your UNIX host and Windows machine.

Downloading 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 either the NetApp Host Agent or 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. You can download the documentation from the Software Description page. For more information on downloading these products, see the Software Download page for SnapManager.
5	Download the software file for your host platform. If you are using the GUI, you must also download that file.

Installing the SnapManager files on a Linux host

Installing the SnapManager files on a Linux host

You install SnapManager on a Solaris host by using the `rpm` command. The program creates the SnapManager installation directory `/opt/NetApp/smo`.

To install the SnapManager files on a Linux host, complete the following steps.

Step	Action
1	Make sure you are logged in as root. Note _____ The installation file is a standard Linux <code>.rpm</code> file.
2	Change to the directory on your Linux host where you put the software you downloaded from the NOW site or copied from the CD.
3	Use the <code>rpm</code> command to install the software. <pre># rpm -ihv <pathname>/netapp.smo.linux-1.1.1.rpm</pre> Result: The command installs the software in the <code>/opt/NetApp/smo</code> directory. Note _____ You must install SnapManager in the default directory, or it will not run. Example: The <code>rpm</code> command installs the software and writes installation information to a log file. <pre># rpm -ivh netapp.smo.linux-1.1.1.rpm Preparing... ##### [100%] 1:netapp.smo ##### [100%] Reminder: You must restart the Host Agent after installing Snap Manager for Oracle #</pre>

Step	Action
4	By default, SnapManager sets the operating system account to bin and the operating system group to root during installation. If you want to change these settings, execute the <code>smo_setup</code> command to supply the user and group that will be running SnapManager. This script also sets the permissions for the directories based on the user and group you supply. Note_____ You should not manually set permissions. _____ When you enter the command line: <pre># /opt/NetApp/smo/bin/smo_setup</pre> SnapManager prompts you for the following information: ◆ The operating system account that will run SnapManager. At this point, the default is oracle. ◆ The operating system group that will run SnapManager. At this point, the default is dba. Example: The following is an example of executing the <code>smo_setup</code> command. <pre># /opt/NetApp/smo/bin/smo_setup SMO post installation configuration: OS account that will run SMO [oracle]: oracle OS group that will run SMO [dba]: dba Reminder: You must restart the Host Agent after installing Snap Manager for Oracle</pre>
5	Complete the installation by adding the repository schema. For information on adding the repository schema, see “Add the repository schema and restart the NetApp Host Agent” on page 49. Then you must configure SnapManager by setting up a profile for your database. For information on setting up a profile, see “Setting up and managing database profiles” on page 67.

Uninstalling the software from an Linux host

If you re-install or install a newer version of SnapManager, you must uninstall the current version. Complete the following steps.

Step	Action
1	Make sure you are logged in as root.
2	Use the <code>rpm</code> command to remove the SnapManager software. Example: The following example uses the <code>rpm</code> command to uninstall SnapManager. <pre># rpm -e netapp.smo</pre>

Installing the SnapManager files on a Solaris host

Installing the software package You install SnapManager on a Solaris host by using the `pkgadd` command. The program creates the SnapManager installation directory `/opt/NTAPsmo`.

To install the file containing the host software, complete the following steps:

Step	Action
1	Make sure you are logged in as root. Note _____ The installation file is a standard Solaris .pkg file. _____
2	Change to the directory on your Solaris host where you put the software you downloaded from the NOW site or copied from the CD.

Step	Action
3	Use the <code>pkgadd</code> command to add the SnapManager packages. Once the packages are added, SnapManager asks if you want to continue installing SnapManager. Answer <code>y</code>. Result: The command installs the software in the <code>/opt/NTAPsmo</code> directory. Note _____ You must install SnapManager in the default directory, or it will not run. _____ Example: This example adds the Solaris packages and installs SnapManager. <pre># pkgadd -d netapp.smo.sunos-1.1.1.pkg all Processing package instance <NTAPsmo> from <./netapp.smo.sunos-1.1.1.pkg> Solaris SnapManager for Oracle (sparc) 1.1.1 Copyright (c) 2006 Network Appliance, Inc. All Rights Reserved. Using </opt/NTAPsmo> as the package base directory. ## Processing package information. ## Processing system information. ## Verifying disk space requirements. ## Checking for conflicts with packages already installed. ## Checking for setuid/setgid programs. This package contains scripts which will be executed with super-user permission during the process of installing this package. Do you want to continue with the installation of <NTAPsmo> [y,n,?] y Installing Solaris SnapManager for Oracle as <NTAPsmo> ## Installing part 1 of 1. /opt/NTAPsmo/bin/schema-oracle.sql /opt/NTAPsmo/bin/smo /opt/NTAPsmo/bin/smo_setup /opt/NTAPsmo/bin/smo_wrapper /opt/NTAPsmo/legal/APACHE_LICENSE-2.0.txt /opt/NTAPsmo/legal/LGPL.txt /opt/NTAPsmo/legal/hibernate-2.1.8.tar.gz /opt/NTAPsmo/legal/jcalendar.zip /opt/NTAPsmo/lib/libcom_netapp_dlm_common_jni_PosixOperatingSystem.so /opt/NTAPsmo/lib/libplugin_dlm.so /opt/NTAPsmo/lib/smo.jar /opt/NTAPsmo/log/smo.log [verifying class <none>] ## Executing postinstall script. Reminder: You must restart the Host Agent after installing Snap Manager for Oracle Installation of <NTAPsmo> was successful.</pre>

Step	Action
4	By default, SnapManager sets the operating system account to bin and the operating system group to root. If you want to change these settings, execute the <code>smo_setup</code> command to supply the user and group that will be running SnapManager. This script also sets the permissions for the directories based on the user and group you supply. Note _____ You should not manually set permissions. _____ When you enter the command line: <pre># /opt/NTAPsmo/bin/smo_setup</pre> SnapManager prompts you for the following information: ◆ The operating system account that will run SnapManager. At this point, the default is oracle. ◆ The operating system group that will run SnapManager. At this point, the default is dba.
5	Complete the installation by doing the following: ◆ Adding the repository schema. ◆ Restarting the NetApp Host Agent. For information on performing these tasks, see “Add the repository schema and restart the NetApp Host Agent” on page 49. Then you must configure SnapManager by setting up a profile for your database. For information on setting up a profile, see “Setting up and managing database profiles” on page 67.

Uninstalling the software from a Solaris host

If you re-install or install a newer version of the software, you must uninstall the current version. Complete the following steps.

Step	Action
1	Make sure you are logged in as root.

Step	Action
2	Use the <code>pkgrm</code> command to remove the SnapManager software <code>NTAPsmo</code>. Answer <code>y</code> when the program asks if you want to remove the SnapManager package. Example: The following example uses the <code>pkgrm</code> command to uninstall SnapManager. <pre># pkgrm NTAPsmo</pre> The following package is currently installed: <pre>NTAPsmo Solaris SnapManager for Oracle (sparc) 1.1.1</pre> Do you want to remove this package? [y,n,?,q] y <pre>## Removing installed package instance <NTAPsmo> ## Verifying package dependencies. ## Processing package information. ## Removing pathnames in class <none> /opt/NTAPsmo/tmp <non-empty directory not removed> /opt/NTAPsmo/profiles /opt/NTAPsmo/mnt /opt/NTAPsmo/log/smo.log /opt/NTAPsmo/log /opt/NTAPsmo/lib/smo.jar /opt/NTAPsmo/lib/libplugin_dlm.so /opt/NTAPsmo/lib/libcom_netapp_dlm_common_jni_PosixOperatingSystem.so /opt/NTAPsmo/lib /opt/NTAPsmo/legal/jcalendar.zip /opt/NTAPsmo/legal/hibernate-2.1.8.tar.gz /opt/NTAPsmo/legal/LGPL.txt /opt/NTAPsmo/legal/APACHE_LICENSE-2.0.txt /opt/NTAPsmo/legal /opt/NTAPsmo/bin/smo_wrapper /opt/NTAPsmo/bin/smo_setup /opt/NTAPsmo/bin/smo /opt/NTAPsmo/bin/schema-oracle.sql /opt/NTAPsmo/bin ## Updating system information.</pre> Removal of <NTAPsmo> was successful.

Add the repository schema and restart the NetApp Host Agent

Completing the installation

To complete the installation, you must add a repository schema to your repository database and restart the Netapp Host Agent. SnapManager creates the schema objects in the default user tablespace for the current user (for example, the database administrator).

Load the schema into the repository database

The SnapManager installation software includes a repository schema.

You must load the SnapManager repository schema into an Oracle database. This can be a database that is already on your system or one that you created for this purpose.

Step	Action
1	Add your schema. Example: This is an example of adding the SnapManager schema to the existing database. This example uses smo_tester as the schema on the database instance called SMOREPO, which is where schema-oracle.sql has to be loaded. (To make this example easier to read, it has been abbreviated by removing most of the Table altered messages.) <pre> oracle \$ export ORACLE_SID=SMOREPO oracle \$ sqlplus smo_tester/netapp@/opt/NetApp/smo/bin/schema-oracle.sql SQL*Plus: Release 9.2.0.1.0 - Production on Fri Jul 29 11:25:57 2005 Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved. Connected to: Oracle9i Enterprise Edition Release 9.2.0.1.0 - 64bit Production With the Partitioning, OLAP and Oracle Data Mining options JServer Release 9.2.0.1.0 - Production Table altered ... SQL> exit Disconnected from Oracle9i Enterprise Edition Release 9.2.0.1.0 - 64bit Production With the Partitioning, OLAP and Oracle Data Mining options JServer Release 9.2.0.1.0 - Production </pre>

Restarting the NetApp Host Agent

Once you install the SnapManager, you must restart the NetApp Host Agent before you can execute SnapManager. Make sure you are logged on as root and enter the following command:

```
ntap_agent start
```

If you need to stop the NetApp Host Agent, use the following command:

```
ntap_agent stop
```

Note

If you reinstall the NetApp Host Agent after you install SnapManager, you must rerun `smo_setup`. This is because SnapManager places a library file in the `/opt/NTAPagent` directory during its installation process. When you uninstall the NetApp Host Agent, it deletes the directory and all of the files in it.

For more information, see the documentation that comes with the NetApp Host Agent.

Install the GUI on a Windows system

Installing the GUI on a Windows system

To install the SnapManager GUI complete the following steps.

Step	Action
1	Download the zip file containing the SnapManager GUI.
2	Extract the zip file to C:\. By default the files are placed in C:\Program Files\NTAPsmo. To execute the GUI, you need to run the smogui.bat file that is placed in this directory. Note_____ Before you extract the files, verify that you have the correct version of the JRE installed. For the most up-to-date list of requirements, see the SnapManager & SnapDrive Compatibility Matrix. This matrix is online on the NOW site at http://now.netapp.com/NOW/knowledge/docs/docs.shtml in the Interoperability section. _____
3	Run the smogui.bat file from the installation directory. This starts the GUI.

Uninstalling the software from a Windows machine

To uninstall the SnapManager GUI from your Windows machine, delete the installation directory. This directory is C:\Program Files\NTAPsmo.

Upgrading to a new version of SnapManager

Upgrading the SnapManager host software

If you are running an earlier version of SnapManager, you may want to upgrade it to the current version.

Note_____

When this document was released, the only versions of SnapManager available were SnapManager 1.1, which runs on a Linux host, and SnapManager 1.1.1, which runs on either a Linux host or a Solaris host. This section explains how to update your Linux host from SnapManager 1.1 to SnapManager 1.1.1.

You do not have to uninstall SnapManager to upgrade it. Instead, you install the latest version of the software on top of the current version.

When you upgrade SnapManager, it keeps your current backups and profiles.

Installing the most recent version of the host software

To install a more recent version of SnapManager, complete the following steps.

Step	Action
1	Ensure that you are logged in as root.
2	Get a copy of the current SnapManager host software and place it on your Linux host. Note_____ You can get a copy of the software by either downloading it from the NOW site or copying it from the CD.

Step	Action
3	Use the <code>rpm -U</code> command to install the software. <pre># rpm -U <pathname>/netapp.smo.linux-1.1.1.rpm</pre> Result: The command installs the software in the <code>/opt/NetApp/smo</code> directory. Note_____ You must install SnapManager in the default directory, or it will not run. _____ Example: The <code>rpm</code> command installs the software and writes installation information to a log file. <pre># rpm -U netapp.smo.linux-1.1.1.rpm</pre> Reminder: You must restart the Host Agent after installing Snap Manager for Oracle <pre>#</pre>

Step	Action
4	Execute the <code>smo_setup</code> command to supply the user and group that will be running SnapManager. This script also sets the permissions for the directories based on the user and group you supply. Note You should not manually set permissions. When you enter the command line: <pre># /opt/NetApp/smo/bin/smo_setup</pre> SnapManager prompts you for the following information: ◆ The operating system account that will run SnapManager. The default is <code>oracle</code>. ◆ The operating system group that will run SnapManager. The default is <code>dba</code>. Example: The following is an example of executing the <code>smo_setup</code> command. <pre># /opt/NetApp/smo/bin/smo_setup SMO post installation configuration: OS account that will run SMO [oracle]: oracle OS group that will run SMO [dba]: dba Reminder: You must restart the Host Agent after installing Snap Manager for Oracle</pre>
5	Restart the NetApp Host Agent. ◆ Halt the NetApp Host Agent by entering the <code>ntap_agent stop</code> command. ◆ Then start it by entering the <code>ntap_agent start</code> command. Example: The following is an example of restarting the NetApp Host Agent. <pre># ntap_agent stop # ntap_agent start</pre>

Step	Action
6	Use the <code>sno version</code> command to confirm that the new version of SnapManager installed correctly. <pre># sno version Snap Manager for Oracle version 1.1.1</pre>

Updating the GUI software

In addition to updating the host software, you can also update the GUI software on your Windows machine. Before you update it, though, you should make a backup. This is because you must remove the GUI software before you can update it.

Installing the most recent version of the GUI software

To install the most recent version of the SnapManager GUI software, complete the following steps.

Step	Action
1	Make a backup of the directory containing information on the SnapManager GUI. In particular, make a backup of C:\Program Files\NTAPsmo\gui\repositories.xml. You can do this from the Windows Explorer or by using a command line prompt. Example: This example uses illustrates how to create a backup from a command line prompt on your Windows machine. <pre> C:\>mkdir smo-backup C:\>mkdir smo-backup\gui C:\>cp "Program Files\NTAPsmo\gui\repositories.xml" smo-backup\gui C:\>dir smo-backup\gui Volume in drive C has no label. Volume Serial Number is 5471-1ACC Directory of C:\smo-backup\gui 05/30/2006 09:37 AM <DIR> . 05/30/2006 09:37 AM <DIR> .. 05/30/2006 09:37 AM 535 repositories.xml 1 File(s) 535 bytes 2 Dir(s) 61,535,858,688 bytes free </pre>
2	Now delete the directory containing the GUI. You can do this by using the Windows Explorer or from a command line prompt. <pre># Delete "C:\Program Files\NTAPsmo\</pre>
3	Get a copy of the new GUI software.
4	Extract the zip file. By default the files are placed in C:\Program Files\NTAPsmo. To execute the GUI, you need to run the smogui.bat file that is placed in this directory.

Step	Action
5	Copy smo-backup\gui\repositories.xml to C:\Program Files\NTAPsmo\gui. You can do this by using the Windows Explorer or from a command line prompt. Example: This example uses a command line prompt to copy the files. <pre> C:\>mkdir "C:\Program Files\NTAPsmo\gui" C:\>cp smo-backup\gui\repositories.xml "C:\Program Files\NTAPsmo\gui" C:\>dir "C:\Program Files\NTAPsmo\gui" Volume in drive C has no label. Volume Serial Number is 5471-1ACC Directory of C:\Program Files\NTAPsmo\gui 05/30/2006 09:45 AM <DIR> . 05/30/2006 09:45 AM <DIR> .. 05/30/2006 09:45 AM 535 repositories.xml 1 File(s) 535 bytes 2 Dir(s) 61,540,352,000 bytes free </pre>
6	Make sure the GUI installed correctly by launching the smogui.bat file. When the GUI starts, you should see your previous default repository in the initial Repository Authentication dialog.

Step	Action
7	Delete the backup of the GUI that you created. You can do this by using the Windows Explorer or from a command line prompt. Example: This example uses a command line prompt to delete the backup directory called smo-backup. <pre>C:\>rm -r smo-backup C:\>dir smo-backup Volume in drive C has no label. Volume Serial Number is 5471-1ACC Directory of C:\ File Not Found</pre>

About this chapter

This chapter provides information about using SnapManager with both the GUI and CLI. It also contains information on setting up a database profile and checking to make sure your environment is correctly set up for SnapManager.

Topics in this chapter

This chapter includes the following topics:

- ◆ [“Getting started with SnapManager”](#) on page 62
- ◆ [“Setting up and managing database profiles”](#) on page 67
- ◆ [“Verify that the system environment is set up correctly”](#) on page 79

Getting started with SnapManager

Getting started

You can execute SnapManager in either of the following ways:

- ◆ From the command line of your UNIX host. See “[Starting SnapManager from the CLI](#)” on page 62.
- ◆ From the GUI on a Windows host attached to your UNIX host. See “[Starting the GUI for SnapManager](#)” on page 63.

You can perform the same operations from both the CLI and the GUI. As a result, the information you need to provide for the operation is essentially the same, regardless of whether you execute the operation from the CLI or the GUI.

In addition, keep in mind the following basic concepts:

- ◆ SnapManager works with profiles you set up. Each profile contains the credentials for the database associated with the profile that enable SnapManager to connect to it and work with it. You must set up a profile for each database you want to use before you can back it up.

Note

SnapManager stores the database credentials in an encrypted file.

- ◆ 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.

Starting SnapManager from the CLI

You can execute SnapManager from the CLI by entering a command at the prompt on your UNIX host. The CLI commands all start with `smo` (SnapManager for Oracle). So you have commands such as `smo check profile_name` or `smo backup`.

For a list of all the SnapManager CLI commands and an explanation of their options and arguments, see “[SnapManager Command Reference](#)” on page 127.

Starting the GUI for SnapManager

To start the SnapManager from the Windows host attached to the UNIX host, run the `smogui.bat` file. When you install the GUI, SnapManager places this file in `C:\Program Files\NTAPsmo`.

Note

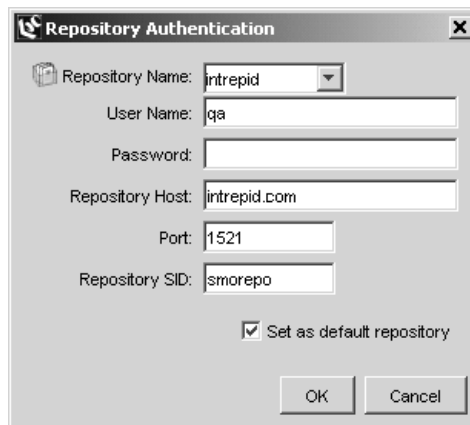
You cannot run the SnapManager CLI on the Windows host.

Authenticating the repository

Before the GUI starts, SnapManager displays a dialog box asking you to authenticate the repository. You must enter the following information:

- ◆ The display name of the repository. You can select this name from a drop-down list of repositories you've set up. To specify a repository that is not in the list, enter the name of that repository here.
- ◆ The username and password for the repository
- ◆ The name of the host where the repository is located
- ◆ The port number to use when accessing the repository (normally this is 1521, Oracle's listener port)
- ◆ The repository SID

If this is the repository you use most of the time, you should also check the box to make it the default. When you do that, SnapManager fills in all the fields except the password field when you start the GUI.



Using the GUI

Once you successfully authenticate the repository the main GUI screen appears.

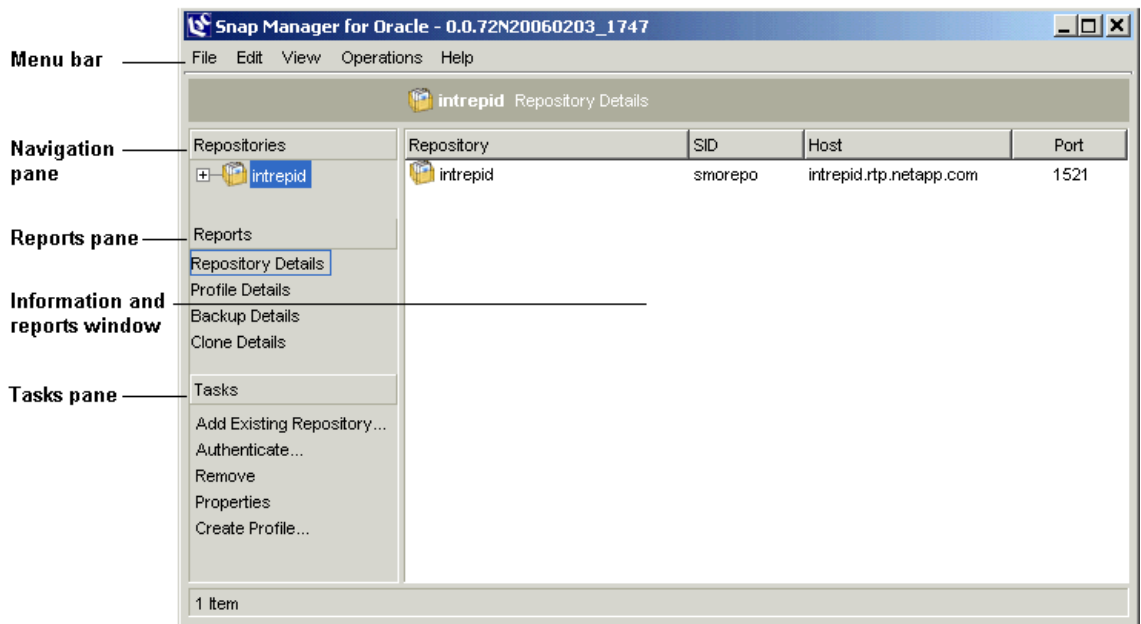
In addition to letting you perform operations, the GUI provides a central method for managing a backup's resources. There are several ways to perform SnapManager operations from the GUI. You can:

- ◆ Select the operation from the pull-down menus in the menu bar. For example, if you wanted to create a profile, you would select Operations > Repository > Create Profile.
- ◆ Click on the operation in the Tasks pane of the GUI screen (see [“Navigating the main GUI screen”](#) on page 64 for a pointer to the task panel).
- ◆ Select the object (repository, profile, or backup) that the operation applies to and right-click on the name. A pop-up menu appears with a list of operations you can perform. Click the operation you want to perform.

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. For other operations, such as verifying that a backup is structurally correct, it displays a pop-up box asking if you want to perform the operation.

Navigating the main GUI screen

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



The Menu bar at the top of the screen lets you perform operations using the pull-down menus.

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. When you click on a repository name, SnapManager displays a list of the profiles contained in the repository. When you click on a profile name, 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 before it displays the backups. You must provide the correct authentication information.

When you want to create a backup, you must first select the repository and then select the profile for the database from that repository. To restore a backup, you must again first select the repository, then chose the profile for the database, and finally, select the backup you want to use. SnapManager lets you retain as many backups as you want, providing you have space for them. For more information on how SnapManager handles retaining backups, see [“SnapManager’s policy for retaining backups”](#) on page 30.

- ◆ **Reports pane.** This pane contains a list of reports. The list of reports changes depending on what you have selected in the Navigation pane. There are

different reports for repositories, profiles, and backups. To see a report, click on the report name. SnapManager displays the report in the main section of the screen.

- ◆ **Tasks pane.** This pane lists the operations you can perform. The list changes depending on what you have selected in the Navigation pane and Reports pane. There are different operations for repositories, profiles, and backups.

When you click on a task, SnapManager takes one of the following actions:

- ❖ It starts the wizard to step you through the operation or
- ❖ It displays a properties box or
- ❖ It displays a pop-up box asking you to confirm the operation.

For example, if you have a repository selected in the Navigation panel, you can choose the task to create a new profile. SnapManager starts the profile wizard.

If you have a backup selected and you choose the task to verify the backup, SnapManager displays a pop-up box asking you to confirm that you want to verify the backup.

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.

Setting up and managing database profiles

SnapManager's profile feature

To simplify working with the databases, SnapManager has you set up a profile for each database. This profile defines all the host and database credentials that SnapManager requires in order to perform the operation you request. SnapManager stores the profile credentials in a repository. It stores information about the repository in an encrypted file on the host.

Note

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

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.

Information required for a profile

Both the CLI command and the GUI require similar information if you want to create a profile. This section takes you through the basic steps. For a syntax example, see [“Creating and managing a profile using the CLI”](#) on page 69. For steps on using the GUI, see [“Creating a profile using the GUI”](#) on page 72.

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

1. Specify a name for the profile. This can be up to 30 characters. It must be unique within the repository that contains the profile.
2. Supply the information required by the repository that contains the profile. You must have already installed the repository before you can create a profile.
 - ❖ 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. You can use either the host name or the IP address.
 - ❖ Specify the port number SnapManager must use to connect to the repository. Normally, this is 1521 (Oracle's default listener port).

3. Supply the information required by the database associated with the profile.
 - ❖ Enter the database name.
 - ❖ Specify the host where the database resides. You can use either the host name or the IP address.
 - ❖ Specify the user name/password required to connect to the database.
 - ❖ Specify the port number SnapManager must use to connect to the database. Normally, this is 1521 (Oracle's default listener port).
 - ❖ If necessary, specify the SID. By default, SnapManager uses the database name as the SID. If you are using RAC, you need to specify the SID you used to when you specified the host information.
4. (Optional) Enter the information required to use the RMAN catalog. Select one of the following two choices:
 - ❖ Enter the -controlfile option if you want to use the control files instead of the RMAN catalog. This option does not have an argument.
 - ❖ If you do not use the -controlfile option, you must provide:
 - * The user name/password required to connect to the RMAN catalog.
 - * The tnsname connection name (this is defined in the tnsname.ora file).
5. Enter the user name/password pair for the NetApp Host Agent.
6. Enter the name of the Oracle database user account (osaccount).
7. (Optional) Enter the information required to use the ASM.
 - ❖ Specify the ASM instance.
 - ❖ Specify the user name/password required to connect to ASM.
 - ❖ Specify the port number SnapManager must use to connect to ASM. Normally, this is 1521 (Oracle's listener port).
 - ❖ If necessary, specify the SID. By default, SnapManager uses the database name as the SID. If you are using RAC, you need to specify the SID you used to when you specified the host information.
8. (Optional) Enter the number of successful backups that you want SnapManager to retain for this database. This option applies only to successful backups that have not been used to create a clone. For more information on how SnapManager retains backups, see "[SnapManager's policy for retaining backups](#)" on page 30.
9. (Optional) Specify how you want SnapManager to handle messages. By default, SnapManager displays the error and warning messages on the console. To change this 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 130.

Authenticating the profile when using it in the GUI

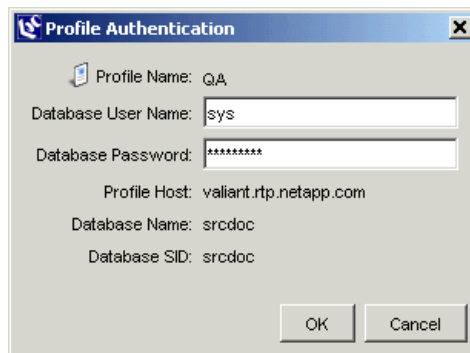
The first time you try to access a profile in a GUI session, SnapManager displays a pop-up asking you to authenticate the profile. Once you authenticate the profile, SnapManager does not ask you for any more information during that session. However, until you authenticate the profile, it does not let you perform

any tasks involving that profile. For example, you cannot create a backup or even view your current backup list, until you authenticate the profile.

SnapManager only requests this information once during a GUI session.

To authenticate the profile, you must supply the following information:

- ◆ The user name for the database
- ◆ The database password



Creating and managing a profile using the CLI

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

```
smo profile create profile_name
{-repository name -username username -password password
  -port port_number -host host [-sid sid]}
{-database name -username username -password password -host host
  -port port_number [-sid sid]}
[-rman [-controlfile | -username username -password password
  -tnsname tnsname] ]
{-hostagent -username username -password password}
{-OSAccount account_name}
[-asm name -port port_number -username username
```

```
-password password [-sid sid] ]
[-retain number]
[-quiet | -verbose]
```

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 154, the “[The smo profile list command](#)” on page 161, and the “[The smo profile show command](#)” on page 162.

Examples of creating profiles: The following command line sets up a profile that uses ASM.

```
# smo profile create TESTER -repository DB1 -host macbeth
  -username bali -password high -port 1521 -database +ASM
  -sid +ASM -username sys -password pacific -port 1521
  -host florida -hostagent -username admin -password netapp
  -OSAccount oracle -asm +ASM -username sys
  -password aloha -port 1521 -host florida -rman -username rman
  -password rmanuser -tnsname DB1
```

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

```
# smo profile create TESTER -repository DB1 -host macbeth
  -username bali -password high -port 1521 -database QA1
  -username sys -password pacific -port 1521
  -host florida -hostagent -username palm -password springs
  -OSAccount oracle -retain 3
```

Changing an existing profile using the CLI

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.

Note

SnapManager verifies the user name/password information before the command completes. If they do not work, the command fails.

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

```
smo profile update -profile profile_name
  [-repository name -host host -username username
    -password password -port port_number [-sid sid] ]
  [-database name -username username -password password -host host
    -port port_number [-sid sid]]
  [-rman [-controlfile | -username username -password password
    -tnsname tnsname] ]
  [-hostagent -username username -password password]
  [-OSAccount account_name]
  [-asm name -username username
    -password password -port port_number [-sid sid] ]
  [-retain number]
  [-remove {asm | rman} ]
  [-quiet | -verbose]
```

For details on the options and arguments used in this command, see “[The smo profile update command](#)” on page 164.

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 MYTEST -database database1
  -username sys -password pacific -host ocean -port 1521
```

This command line updates the repository information for the profile MYTEST.

```
# smo profile update MYTEST -repository database2 -host shakespeare
  -username sys -password venice -port 1521
```

Deleting a profile using the CLI

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

```
smo profile delete profile_name [-force] [-quite] [-verbose]
```

For details on the options and arguments used in this command, see “[The smo profile delete command](#)” on page 159

Example: The following command line deletes the profile MYTEST. It includes the `-force` option, which tells SnapManager to remove any backups associated with that profile.

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

Creating a profile using the GUI

To create a profile from the GUI, complete the following steps.

Step	Action
1	From the Navigation pane, select the repository you want associated with this profile.
2	Specify the Create Profile operation by selecting Create Profile from: ◆ The pop-up menu box that appears when you right-click on the repository name. ◆ The Tasks pane. ◆ The main menu bar: Operations > Repository > Create Profile. The SnapManager for Oracle Profile Wizard starts.
3	At the Profile Settings screen, enter the following information: ◆ The profile name. You can give it any name you choose. ◆ The name of the Oracle operating system account to use on the host. This example uses oracle. ◆ The number of successful backups to retain. Click Next to continue.

SnapManager for Oracle Profile Wizard

Profile Settings

Name for the profile.

Profile Name:

Oracle O/S account to use on the host.

Host Account:

Backups to Retain:

< Back Next > Cancel

Step	Action
4	At the Database Settings screen, enter the following information: ◆ The name and password of your account that has SYSDBA privileges. ◆ The host where the database resides. ◆ The database name. ◆ The database SID if you want to use RAC support. If you do not provide a SID, SnapManager uses the database name. ◆ The port number. Normally this is 1521, Oracle's listener port. Click Next to continue.
5	At the NetApp Host Agent Settings screen, enter the following information: ◆ The user name for the account. The default name when you install this product is admin. ◆ The password for the account. The default password when you install this product is netapp. Click Next to continue.

SnapManager for Oracle Profile Wizard

Database Settings

SYSDBA Privileged User Name:

Password:

Host where the database resides:

Host:

Database Name:

Defaults to database name if not specified.

Database SID:

Port:

< Back Next > Cancel

SnapManager for Oracle Profile Wizard

NetApp Host Agent Settings

Host Agent admin user name.

User Name:

Host Agent password.

Password:

< Back Next > Cancel

Step	Action
6	Skip the ASM Settings screen if you do not use ASM. To enable SnapManager to work with ASM, click the box Use ASM and provide the following information in the ASM Credentials section: ◆ The port number for connection with ASM. Normally, this is 1521, Oracle's listener port. ◆ The user name and password for connection to the Oracle database. ◆ The Oracle SID for the ASM instance. Click Next to continue.

SnapManager for Oracle Profile Wizard

ASM Settings 

☐ Use ASM

ASM Credentials

Port:

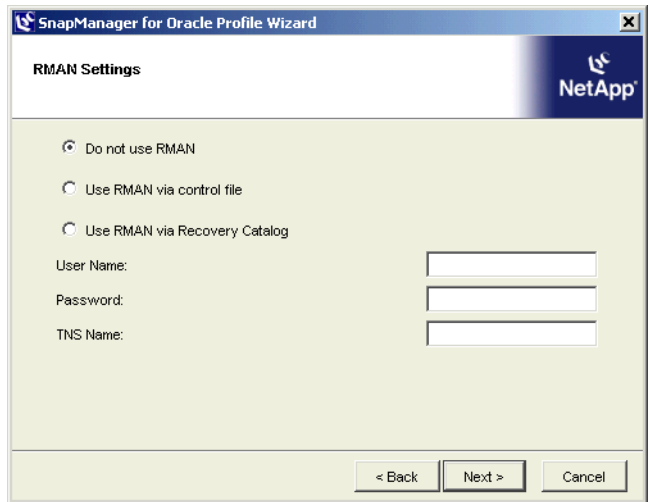
User Name:

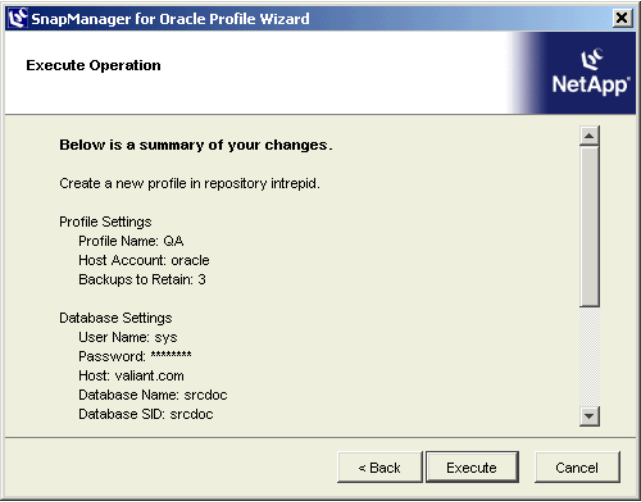
Password:

ASM SID:

< Back Next > Cancel

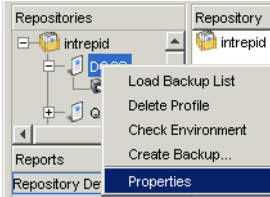
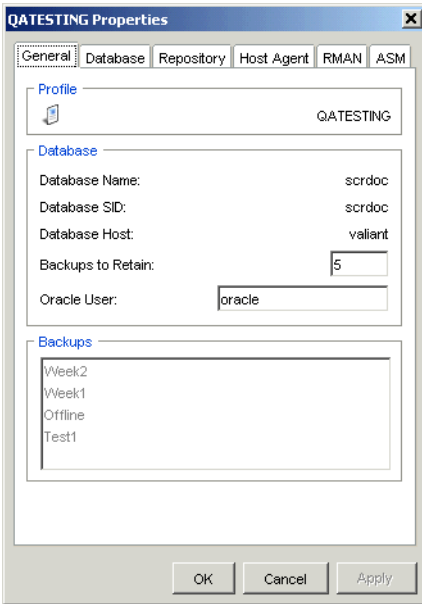
Step	Action
7	At the RMAN Settings screen, click the box “Do not use RMAN” if you don’t want to use it. To use RMAN, deselect the box and enter the following information: ◆ Click the box “Use RMAN via the control file” if you want to use the control files instead of using an RMAN recovery catalog. ◆ Click the box “Use RMAN via Recovery Catalog” and supply the Oracle Call Interface (OCI) connection information. ◆ Enter the username and password for connecting to the RMAN catalog. ◆ The tnsname connection name (this is defined in the tnsname.ora file). Click Next to continue.



Step	Action
8	At the Execute Operation screen, verify the information you've supplied. If you need to correct anything, click Back until you reach the screen containing the information you want to correct. If the information is correct and you want to create this profile, click Execute. 
9	The Completing the SnapManager for Oracle Profile Wizard screen appears. This screen displays the results of the profile creation process. It also includes a tab called Operation Details. Select Operation Details to see information about the profile creation operation. If you successfully created the profile, you can click Finish or you can click Operation Details to see information about the profile. If the profile creation failed, you can click Back to return to screens and correct information. 

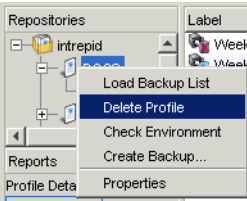
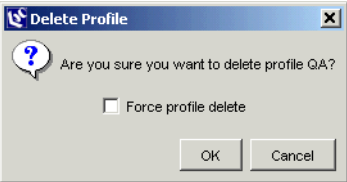
Changing an existing profile using the GUI

To modify a profile from the GUI, complete the following steps.

Step	Action
1	Select the profile you want to modify. You can select the profile from the Navigation pane by expanding repository that contains the profile.
2	Right-click on the profile and select Properties from the pop-up menu. 
3	The Properties dialog box lets you change the following: ◆ General information about the profile ◆ Database information ◆ Repository information ◆ NetApp Host Agent information ◆ RMAN information ◆ ASM information Click the tab for the section you want to update and make your changes. You can modify settings on multiple tabs. 
4	Click OK when you've finished modifying the profile.

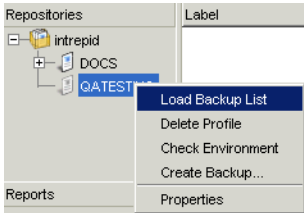
Deleting a profile using the GUI

To delete a profile using the GUI, complete the following steps.

Step	Action
1	Select the profile you want to delete. You can select the profile from the Navigation pane by expanding repository.
2	Right-click on the profile you want to delete and select Delete Profile from the pop-up menu. 
3	The Delete Profile pop-up box appears. You can check the “Force profile delete” if you want SnapManager to force the deletion. If the profile has backups associated with it, the delete fails unless you check this box. When you check it, SnapManager deletes all backups for that profile. Click OK to confirm the deletion. 

Displaying a list of backups associated with a profile

To display a list of backups associated with a profile in the GUI, select the Load Backup List option, which is available in both the menus and the Tasks pane.



You can also expand the view of the profile in to display the backups that are associated with it.

Verify that the system environment is set up correctly

Confirm the environment for SnapManager

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

- ◆ SQLPlus and RMAN can be run from this account.
- ◆ All environment variables required by SnapManager are set correctly.

In addition, if you supply the name of a profile, SnapManager makes sure that

- ◆ The database and the storage systems are accessible.
- ◆ The RMAN catalog is available (this only needs to be available if you are using RMAN).

Verifying the environment for your system from the CLI

To check the system environment for a profile from the CLI, use the `smo check` command. It has the following general format:

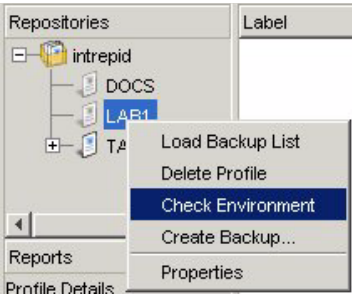
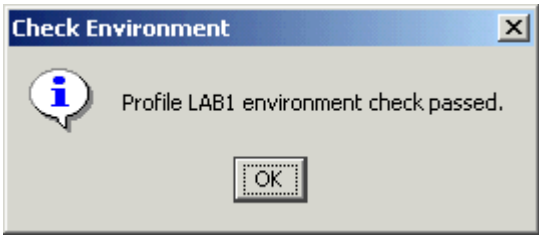
```
smo check -profile profile_name [-quiet] [-verbose]
```

For details on the options and arguments used in this command, see “[The smo check command](#)” on page 136.

Verifying the environment from the GUI

To verify that the system has been set up correctly from the GUI, complete the following steps.

Step	Action
1	Select the profile for which you want to check the environment. You can select the profile from the Navigation pane by expanding repository that contains the profile.

Step	Action
2	Right-click on the profile you are interested in and select Check Environment from the pop-up menu.  The screenshot shows a tree view under 'Repositories' with 'intrepid' expanded. Under 'intrepid', there are 'DOCS' and 'LAB1' (highlighted). A right-click context menu is open over 'LAB1', showing options: 'Load Backup List', 'Delete Profile', 'Check Environment' (highlighted), 'Create Backup...', and 'Properties'. At the bottom of the interface, there are tabs for 'Reports' and 'Profile Details'.
3	SnapManager displays pop-up box telling you whether the check passed or failed. Click OK.  The screenshot shows a dialog box titled 'Check Environment' with a close button (X) in the top right corner. Inside the dialog, there is an information icon (i) and the text 'Profile LAB1 environment check passed.' At the bottom center, there is an 'OK' button.

About this chapter This chapter describes how to use SnapManager to back up and restore databases. It also explains some of the other tasks you can perform with SnapManager.

Topics in this chapter This chapter includes the following topics:

- ◆ “[Creating database backups](#)” on page 82
- ◆ “[Verifying a backup](#)” on page 88
- ◆ “[Restoring and recovering a database](#)” on page 90

Creating database backups

Using the SnapManager backup feature

SnapManager lets you create backups of entire databases and portions of the databases. 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 21.

General steps for backing up data using the CLI or GUI

Both the CLI command and the GUI wizard require similar information if you want to backup a database. This section takes you through the basic steps. For a syntax example, see [“Creating a backup from the CLI”](#) on page 84. For steps on using the GUI, see [“Creating a backup from the GUI”](#) on page 84.

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 (online), the Oracle status is OPEN, and archive log mode is enabled.
 - ❖ Offline. The database is mounted, but not available to users (offline), and 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 SnapManager `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 comma-separated list without spaces.

Note

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

- ❖ (Optional) Whether you want to enter a label (i.e., an easily identified name) for the backup. The label name must be unique within that profile. You can use alphanumeric characters, underscores, and hyphens as long as the first character in the name is not a hyphen.

Tip: To make identifying the backup easier, NetApp recommends that you supply a label for this backup.

If you do not supply a label, SnapManager creates a default label that has the format `<scope>_<type>_<date>` where

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

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

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

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

- ❖ (Optional) Whether you want to enter a comment that describes the backup. For example, you might enter 'Production test' so that you can quickly identify the contents of the backup.

Note

If you are executing this command from the CLI, you may need to use backslashes (\) to escape the single quote. Some shells automatically strip out single quotes.

4. (Optional) Enter the `-force` option if you want SnapManager to force the database to change its state if it is not in the state you requested for the backup operation.
5. (Optional) Specify how you want SnapManager to handle messages when executing the command from the CLI. For more information on these options, see [“Common options for handling messages shown on the console”](#) on page 130.

Creating a backup from the CLI

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

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

For details on the options and arguments used in this command, see “[The smo backup command](#)” on page 131.

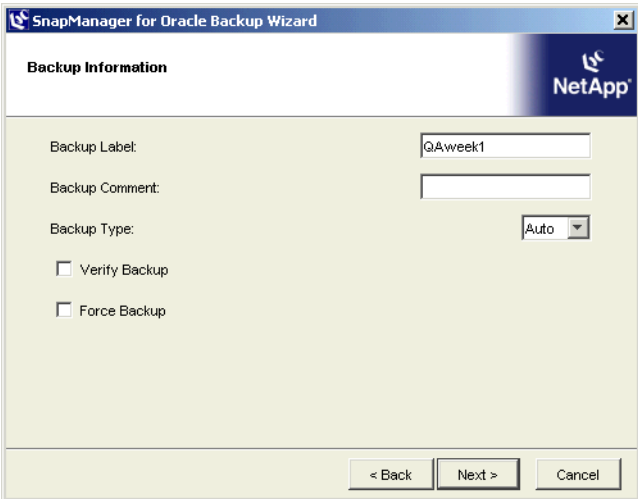
Example: The following command line creates backup with the label `mytestbed`.

```
# smo backup -auto -full -profile MYTEST -label mytestbed
```

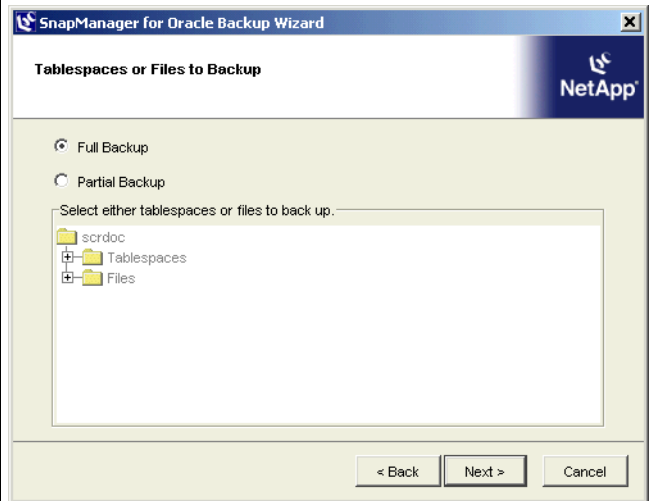
Creating a backup from the GUI

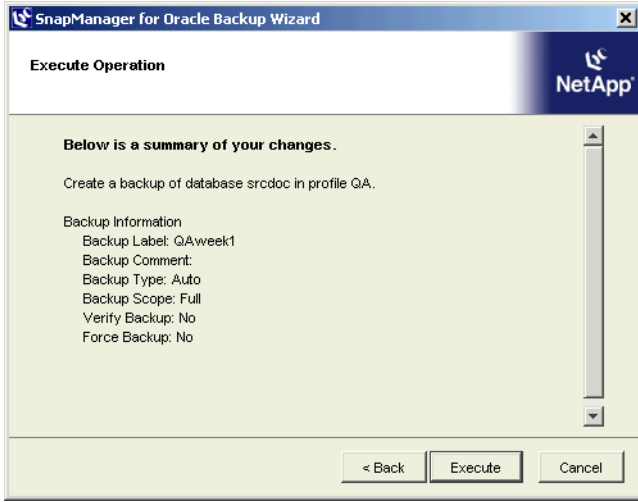
To create a backup using the SnapManager for Oracle Backup Wizard provided by the GUI, complete the following steps.

Step	Action
1	From the Navigation panel, select the profile containing the database you want to back up.
2	Specify the Create Backup operation by selecting Create Backup from: ◆ The pop-up menu box that appears when you right-click on the profile name. ◆ The Tasks pane. ◆ The main menu bar: Operations > Profile > Create Backup. The SnapManager for Oracle Backup Wizard starts.

Step	Action
3 At the Backup Information screen, enter the following information: ◆ (Optional) A label you can use to identify the backup. NetApp recommends you do this so that you can easily identify the backup. If you do not specify a label, then SnapManager creates a default backup label that includes information on the type of backup (Full or Partial), whether database was offline or online (Cold or Hot), and the date and time when the backup was made. ◆ (Optional) A comment describing the backup.	
◆ The type of backup you want to make. Select the type from the drop-down list. If you select Auto, SnapManager determines whether to perform an online or offline backup based on the following: ❖ If the archive log mode is enabled and the database is in an OPEN state, 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. ◆ (Optional) Check the box next to Verify backup if you want SnapManager to confirm that the backup is in a valid format for Oracle. This operation checks the block structure. If you don't want to perform a verify operation at this time, you can execute it later as a separate operation. ◆ (Optional) Check the box next to Force if you want SnapManager to force the state of the database into the specified backup mode. For example, SnapManager might change the state of the database from open to mounted. Click Next to continue.	

Step	Action
4	At the Tablespaces or Files screen, choose the type of backup you want. Choose one of the following two options: ◆ Click the radio button next to Full backup if you want SnapManager to make a backup of the entire database. ◆ Click the radio button next to Partial backup if you want SnapManager to make a backup of either tablespaces or specific data files. If you select this option, you must also use the window below it to indicate which tablespaces or which data files need to be backed up. Note _____ You cannot select both tablespaces and data files. You must choose one or the other. _____ Click Next to continue.



Step	Action
5	At the Execute Operation screen, verify the information you've supplied. If you need to correct anything, click Back until you reach the screen containing the information you want to correct. If the information is correct and you want to create this backup, click Execute. 
6	The Completing the SnapManager for Oracle Backup Wizard screen appears. This screen displays the results of the backup creation process. It also includes a tab called Operation Details. Select Operation Details to see information about the backup creation operation. If you successfully created the backup, you can click Finish or you can click Operation Details to see information about the profile. If the backup failed, you can click Back to return to screens and correct the information. 

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. You can have SnapManager perform the verification immediately after making the backup, or you can execute the verify operation at a later time.

General steps for verifying a backup from the CLI or GUI

Both the CLI command and the GUI require similar information if you want to verify a backup. This section takes you through the basic steps. For a syntax example, see [“Verifying a backup from the CLI”](#) on page 88. For steps on using the GUI, see [“Verifying a backup using the GUI”](#) on page 89.

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 128.)
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 130.

Verifying a backup from the CLI

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

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

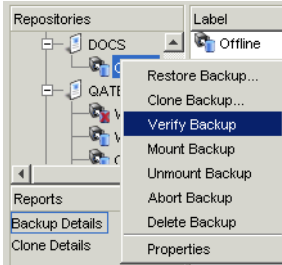
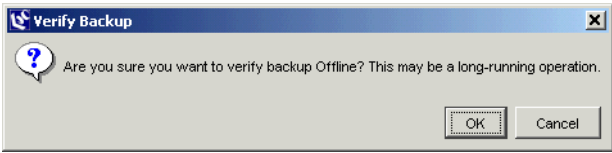
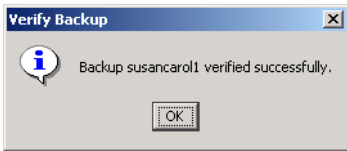
For details on the options and arguments used in this command, see [“Using the `smo verify` command”](#) on page 173.

Example: The following command line verifies the backup mytestbed.

```
# smo verify -label mytestbed -profile MYTEST
```

Verifying a backup using the GUI

To verify a backup from the SnapManager GUI, complete the following steps.

Step	Action
1	From the Navigation panel, select the profile containing the backup you want to verify. From the Navigation panel, select the backup you want to verify. To get to the backup, choose Repository > Profile.
2	Select the Verify Backup operation. 
3	SnapManager displays a pop-up window warning you that the verify operation could take a long time. Click yes if you want to continue with the verify operation. 
4	When the operation finishes, SnapManager displays a pop-up window letting you know whether the verification operation succeeded or failed. Click OK. 
5	To see the results of the verify operation, open the Property page for the backup. You can do this by right-clicking on the backup name and selecting Properties from the pop-up menu.

Restoring and recovering a database

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 restore` operation also gives you the option of deciding how much of the backup you want 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 (i.e., the last commit).
- ◆ A specific date and time.
- ◆ A specific Oracle System Change Number (SCN).

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 21.

General steps for restoring and recovering a backup from the CLI or GUI

Both the CLI command and the GUI wizard require similar information if you want to restore and recover a backup. This section takes you through the basic steps. For a syntax example, see [“Restoring a backup from the CLI”](#) on page 92. For steps on using the GUI, see [“Restoring a backup from the GUI”](#) on page 92.

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 128.)
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
 - ❖ Specific data file
 - ❖ Control files
4. (Optional) Specify that you want to recover the database to:
 - ❖ The last transaction
 - ❖ An SCN (use only if you are doing a point-in-time recovery of the backup)
 - ❖ A specific date and time (use only if you are doing a point-in-time recovery of the backup)
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.
6. (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 130.

Restoring a backup from the CLI

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

```
smo restore {-id guid | -label label_name}
            -profile profile_name
            {-complete | -files list | -tablespaces list | -controlfiles}
            [-recover [-when {auto | scn | date }]]
            [-force] [-quiet | -verbose]
```

Note—
The default value for the `-recover` option is the `auto` argument. So, if you do not specify a value with it, SnapManager performs the recovery as if you had supplied the `auto` argument.

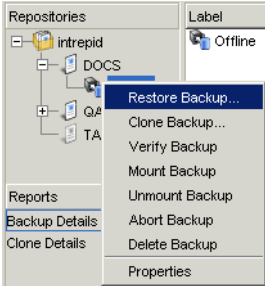
For details on the options and arguments used in this command, see “[The smo restore and recovery command](#)” on page 169.

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

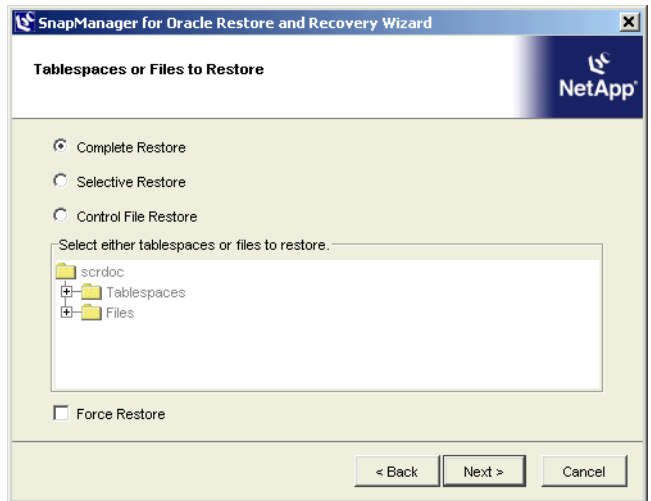
```
smo restore -complete -label ProdFriday -profile Production
-recover -when auto
```

Restoring a backup from the GUI

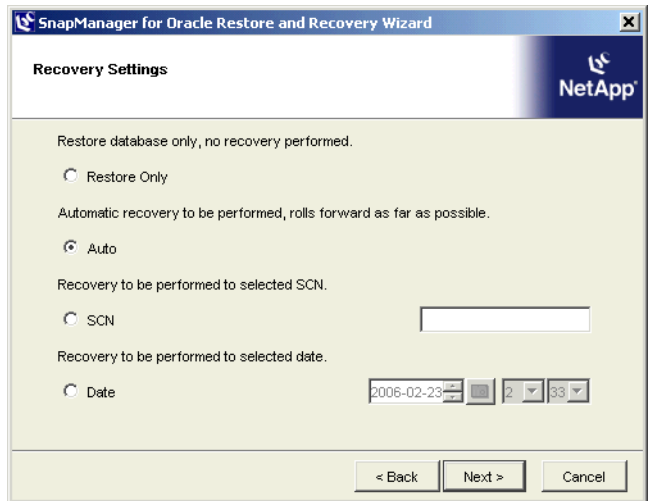
To restore a backup using the SnapManager for Oracle Restore and Recovery Wizard provided by the GUI, complete the following steps.

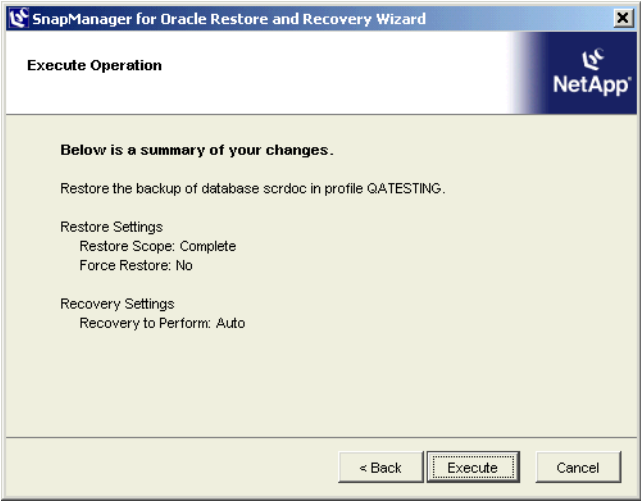
Step	Action
1	From the Navigation pane, select the profile containing the backup you want to restore.
2	Select the backup.
3	Select the Restore Backup operation. The SnapManager for Oracle Restore and Recovery Wizard starts. 

Step	Action
4	At the Tablespaces or Files to Restore screen, specify the type of restore you want to perform. ◆ Click the radio button next to Complete Restore if you want SnapManager to restore the entire backup. ◆ Click the radio button next to Selective Restore if you only want to restore certain tablespaces and files. If you select this option, you must also use the window below it to indicate which tablespaces or data files you want to restore. ◆ Click the radio button next to Control File Restore if you want to restore only the control files. Note— For more information on these types of restore operations, see “Approaches to backing up, restoring, and recovering data” on page 21 Click Next to continue.



Step	Action
5	At the Recovery Settings screen, select the radio button next to Restore only if you do not want SnapManager to restore the database. If you want SnapManager to restore the database, select the type of recovery: ◆ Click the radio button next to Auto if you want SnapManager to recovers the database to the last transaction or commit. ◆ Click the radio button next to SCN and enter an SCN in the box. With this option SnapManager rolls forward the Oracle SCN log files until it reaches the SCN number you specify. ◆ Click the radio button next to Date and supply a date and time. SnapManager recovers the database up to the date and time you specify. Note_____ For more information on these types of recovery operations, see “Approaches to backing up, restoring, and recovering data” on page 21 Click Next to continue.



Step	Action
6	At the Execute Operation screen, verify the information you've supplied. If you need to correct anything, click Back until you reach the screen containing the information you want to correct. If the information is correct and you want to complete this operation, click Execute. 
7	The Completing the SnapManager for Oracle Restore and Recovery Wizard screen appears. Click the tab Operation Details to see details. If the operation completed successfully, click Finish. If the operation failed, you can click Back to return to screens and correct the information.

About this chapter

This chapter describes how to use SnapManager to manage information about the databases.

Topics in this chapter

This chapter includes the following topics:

- ◆ “[Aborting an operation](#)” on page 98
- ◆ “[Displaying information about backups](#)” on page 101
- ◆ “[Removing backups](#)” on page 104
- ◆ “[Mounting and unmounting backups](#)” on page 107
- ◆ “[Checking the version of SnapManager](#)” on page 111
- ◆ “[Managing log files](#)” on page 112

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 an abort operation to release the resources associated with that operation.

In addition, SnapManager may have performed an operation on the database or storage that needs to be undone; for example, taking a snapshot. You should use the abort operation to undo the action.

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

Using the abort command

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

- ◆ Clean up after an incomplete backup

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

For more information SnapManager's states, such as an incomplete backup, see "[SnapManager operational states](#)" on page 32.

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

When you abort a backup, SnapManager attempts the following:

- ❖ To kill the SnapManager processes

- ❖ To cleanly abort any pending SnapManager backup operations on the target database and/or on the storage systems.
- ❖ To unlock the target database for other SnapManager operations. For example, suppose that the backup operation put a tablespace in backup mode. The `smo manage abort` command takes the tablespace out of backup mode. If a snapshot was created during the backup operation, the `smo manage abort` command removes that snapshot from the storage system.

There are two important cautions you need 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. For an example, you may need to take a tablespace out of backup mode, remove a useless snapshot, or remove the target database lock file.

General steps for aborting an operation from the CLI or the GUI

Both the CLI command and the GUI require similar information if you want to abort an operation. This section takes you through the basic steps. For a syntax example, see [“Aborting an operation from the CLI”](#) on page 100. For steps on using the GUI, see [“Aborting an operation from the GUI”](#) on page 100.

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 128.)
2. Specify the name of the profile associated with the operation.
3. (Optional) Specify whether you want SnapManager to force the abort to happen.
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 130.

Aborting an operation from the CLI

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

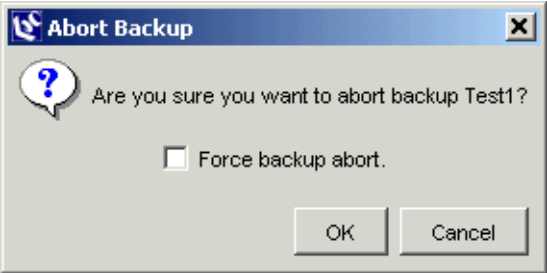
```
smo manage abort {-id guid |-label label_name}
                 -profile profile_name [-force] [-quiet | -verbose]
```

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

Aborting an operation from the GUI

To abort a backup operation from the SnapManager GUI, complete the following steps.

Note _____
Currently, you can only use the abort operation in the GUI to abort backup operations.

Step	Action
1	Select Abort. You can select this option only if the backup operation is no longer running.
2	SnapManager displays a pop-up window asking you to confirm that you want to abort the backup operation. If you want SnapManager to perform the abort even if resources are still tied up by the backup operation, check the “Force backup abort” box. Click OK to execute the abort operation. 

Displaying information about backups

Listing information in current backups

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

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

General steps for listing backup information from the CLI or GUI

Both the CLI command and the GUI require similar information if you want to display information about backups. This section takes you through the basic steps. For a syntax example, see “[Listing backup information from the CLI](#)” on page 102. For steps on using the GUI, see “[Listing backup information from the GUI](#)” on page 102.

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

1. Specify the name of the profile from which the backups you want to view were created.
2. (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 130.

Listing backup information from the CLI

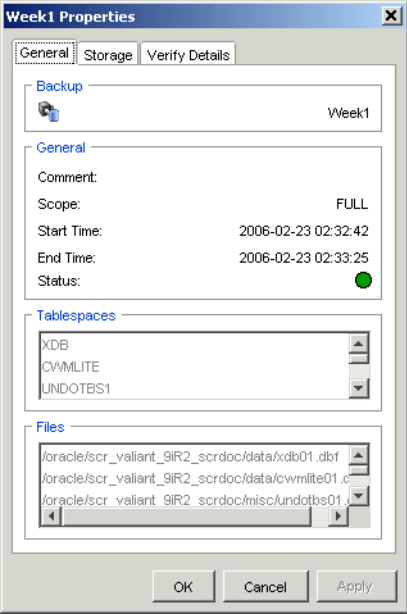
To display information on a backup from the CLI, use the `smo manage list` command. It has the following general format:

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

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

Listing backup information from the GUI

To display information about a backup from the GUI, look at the Property dialog for that backup by completing the following steps.

Step	Action
1	Select the profile you want to modify. You can select the profile from the Navigation pane by expanding repository that contains the profile.
2	Right-click on the backup and select Properties from the pop-up menu.
3	The Properties dialog box provides information about the backup. It organizes the information into the following sections: ◆ General information about the backup ◆ Storage information ◆ Information from the verify operation, if it was performed Click the tab for the section you want to view. 

Step	Action
4	Click OK when you've finished.

Removing backups

Methods for removing backups

You can remove backups for a database at any time. It's a good practice to remove a backup when you no longer need it. For example, Data ONTAP has a limit of 255 snapshots per volume. If you remove backups you no longer need, you reduce the chance of reaching that limit.

There are three ways to remove SnapManager backups:

- ◆ Use the SnapManager remove operation to remove a specific backup.
- ◆ Use the SnapManager retain policy to allow the backups to expire.

Note

SnapManager uses a sliding window approach to retaining backups. It keeps successful backups based on the value you supply for the retain option when you create or update a profile. 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 30.

- ◆ Use the SnapManager profile delete operation to remove the entire profile, including the backups associated with it. (For more information, see [“Deleting a profile using the CLI”](#) on page 71.)

Using the remove backups feature

SnapManager’s remove operation lets you specify which backup 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.

General steps for removing a backup from the CLI or GUI

Both the CLI command and the GUI require similar information if you want to remove a backup. This section takes you through the basic steps. For a syntax example, see [“Removing a backup using the CLI”](#) on page 105. For steps on using the GUI, see [“Removing a backup using the GUI”](#) on page 105.

The basic steps for removing a backup are as follows:

1. Specify either the name of the backup you want to remove or the number of backups you want to keep.

The name of the backup can be either:

- ❖ The label you supplied when you created the backup
 - ❖ The GUID.
2. Specify whether you want SnapManager to retain any of the backups. If you set the `-retain` option to 0, SnapManager removes all the backups.
 3. Specify the name of the profile from which the backup was created.
 4. (Optional) Specify whether you want SnapManager to force the removal of the backup.
 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 130.

Removing a backup using the CLI

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

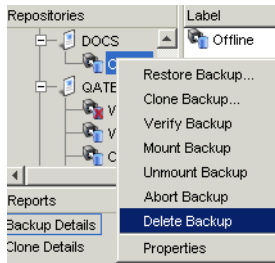
```
smo manage remove {-label label_name | -id guid |
  -retain number} -profile profile_name [-force]
  [-quiet | -verbose]
```

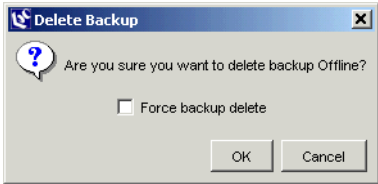
For details on the options and arguments used in this command, see “[The smo manage remove backup command](#)” on page 147.

Removing a backup using the GUI

To remove a backup using the GUI, complete the following steps.

Step	Action
1	Select the backup you want to remove.
2	Right-click on the backup you want to delete and select Delete Backup from the pop-up menu.



Step	Action
3	The Delete Backup pop-up box appears. You can check the “Force backup delete” if you want SnapManager to force the deletion. When you check this box, SnapManager deletes the backup even if there are problems freeing the resources associated with the backup. For example, if the backup was cataloged with RMAN, but the RMAN database no longer exists, including <code>-force</code> on the command line lets SnapManager delete the backup even though it cannot connect with RMAN Click OK to confirm the deletion. 

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

The `smo manage 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 from the CLI or GUI

Both the CLI command and the GUI require similar information if you want to mount or unmount a backup. This section takes you through the basic steps. For a syntax example, see “[Mounting and unmounting a backup from the CLI](#)” on page 108. For steps on using the GUI, see “[Mounting a backup using the GUI](#)” on page 109 and “[Unmounting a backup using the GUI](#)” on page 109.

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 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 128.)
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 130.

Mounting and unmounting a backup from the CLI

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

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

For details on the options and arguments used in this command, see “[The smo manage mount command](#)” on page 145.

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

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

For details on the options and arguments used in this command, see “[The smo manage unmount command](#)” on page 150.

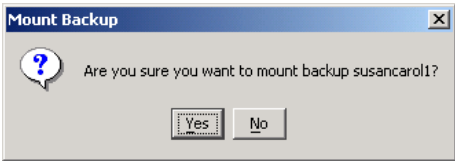
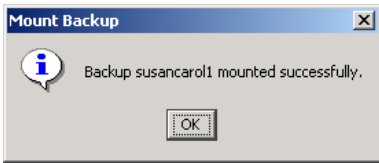
The following is an example of a mount operation.

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

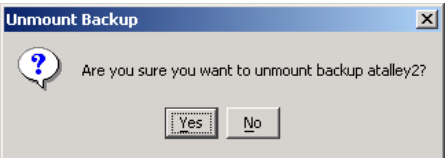
The following is an example of an unmount operation:

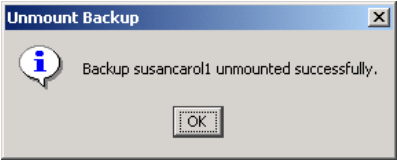
```
# smo manage unmount -label test -profile MYTEST -verbose  
SMO-13039 [INFO ]: Operation GUID: 8abc01b308449e7d0108449e94f80001  
SMO-08027 [INFO ]: Beginning to disconnect connected mount(s)  
[/u02/app/oracle/esdb1_SMO_ESDB1_P_C_20051219142901EST_8abc01b3084  
47d850108447dc8bb0002].  
SMO-08029 [INFO ]: Done disconnecting connected mount(s)  
[/u02/app/oracle/esdb1_SMO_ESDB1_P_C_20051219142901EST_8abc01b3084  
47d850108447dc8bb0002].  
SMO-13011 [INFO ]: Successfully completed operation: Manage  
SMO-13036 [INFO ]: Operation Status: SUCCESS  
SMO-13037 [INFO ]: Elapsed Time: 0:00:32.435
```

Mounting a backup using the GUI To mount backup from the SnapManager GUI, complete the following steps.

Step	Action
1	From the Navigation panel, select the backup you want to mount.
2	Select the Mount backup operation.
3	SnapManager displays a pop-up box asking you to confirm that you want to mount the backup. Click yes. 
4	When the operation finishes, SnapManager displays a pop-up box telling you whether the operation succeeded or failed. Click OK. 

Unmounting a backup using the GUI To unmount backup from the SnapManager GUI, complete the following steps.

Step	Action
1	From the Navigation panel, select the backup you want to unmount.
2	Select the Unmount backup operation.
3	SnapManager displays a pop-up box asking you to confirm that you want to unmount the backup. Click yes. 

Step	Action
4 When the operation finishes, SnapManager displays a pop-up box telling you whether the operation succeeded or failed. Click OK.	

Checking the version of SnapManager

Using the version operation

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

Determining the version of SnapManager

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

```
smo version [-verbose]
```

Managing log files

Working with SnapManager's log files

SnapManager creates log files containing information about all its operations. Each log file can contain information about multiple operations. You can use this information if you have a problem with SnapManager.

By default SnapManager places the log files in `/opt/NetApp/smo/log` on Linux hosts and `/opt/NTAPsmo/log` on Solaris hosts.

Automatic log file rotation

SnapManager keeps a maximum of 12 log files. When a log file reaches 10MB, SnapManager saves the file and starts a new one. Once it has 12 log files, it deletes the oldest log file and starts a new cycle of log files.

SnapManager names the files `smo.log`, `smo.log.0`, up to `smo.log.10`.

About this chapter

This chapter discusses using SnapManager to clone backups of database instances. It also presents the steps you need to follow to clone a database from either the CLI or the SnapManager GUI.

Topics in this chapter

This chapter includes the following topics:

- ◆ [“General information about cloning backups using SnapManager”](#) on page 114
- ◆ [“Creating a clone map file”](#) on page 116
- ◆ [“Cloning a database”](#) on page 120
- ◆ [“Using cloning with RAC databases”](#) on page 125

General information about cloning backups using SnapManager

SnapManager's clone feature

SnapManager's clone feature lets you quickly create a copy of a database instance that is an exact replica of the original database. You create the clone from a full backup that was created while the database was offline.

Using SnapManager to create the clone provides the following advantages:

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

Note

You must create a clone map file before you execute the `smo clone` command or the clone wizard. For instructions on creating this file, see [“Creating a clone map file”](#) on page 116.

How the clone feature works

SnapManager creates a clone using a snapshot of the database. This differs from a traditional clone, which creates a physical copy of the entire original database instance. To create a physical copy, you must have space available that is equal to the current size of the database instance. For example, if your database instance is 17MB, you would need a minimum of an additional 17MB available to hold the copy. With SnapManager, you can create very space efficient clones very quickly.

SnapManager's clone operation uses the FlexClone feature available with Data ONTAP (see the Data ONTAP documentation for more information on FlexClone). Instead of creating a physical copy of the database, SnapManager takes snapshots of the database. The snapshot is a logical copy of the database, so it is quick to make and does not take up much space.

The basic steps for creating a clone are as follows:

1. Create a full backup of the database instance while it is offline.
2. Create a map file for the database instance. SnapManager uses this file to correctly map information in the clone based on the location of this information in the original database instance
3. Use either the CLI command or GUI Clone Wizard to create the clone.

Key points about cloning

For this release, SnapManager only supports cloning from a full, offline backup. In addition, SnapManager only supports cloning a database on the same host. You must give the clone a new Oracle System Identifier (SID). Oracle does not permit you to run two databases with the same SID simultaneously on the same host.

As part of the cloning process, SnapManager creates the necessary Oracle files (such as `init<clone_SID>.ora`) and parameters for the cloned database.

When SnapManager clones the storage for a database, it changes the file system mount point, but it does not change the directory structure under the mount point.

You can clone a Real Application Cluster (RAC) database instance as well as a non-clustered database instance. By default, a RAC clone starts as a single database instance. For more information, see [“Using cloning with RAC databases”](#) on page 125.

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

Files created during the clone process

When you clone a database, SnapManager creates a new `init<clone_SID>.ora` file for the database in the `$ORACLE_HOME/dbs` directory.

In addition, if the database you cloned was using an spfile, then SnapManager creates an spfile for the clone. It places this file, called `spfile<clone_SID>.ora` in the `$ORACLE_HOME/dbs` directory as well. It also creates the `bdump`, `cdump`, and `udump` directories.

Creating a clone map file

Using up a clone map file

Before you can create a clone, you must set up a file that contains the mapping information from the source database instance to the cloned database instance. SnapManager uses this information to determine where to place the files it clones.

Setting up a clone map file

You can create a clone map file with any text editor. The clone map file contains key-value pairs that show the source locations of the files and the destination locations.

When you set up a clone map file, use the following conventions:

- ◆ Enter the key-value pairs in the format:

`key=value`

For a list of keys you can use in the clone map file, see the table in “[Key-value pairs accepted in the clone map file](#)” on page 117. In this table the term *mapping* refers to a string in the format `source => destination`.

- ◆ Place each key-value pair on a separate line.
- ◆ Use a colon to delimit multiple mappings or entries per key.
- ◆ Use the # to indicate comment lines. SnapManager ignores all lines that start with #.
- ◆ (Recommended) Use an extension of `.clonemap` for the file. SnapManager accepts any extension for this file, but NetApp recommends you use `.clonemap` so that you can quickly locate the file.

**Key-value pairs
accepted in the
clone map file**

SnapManager lets you use the following keys in this file.

Key	Value
data	Colon-delimited mappings of the data files. These are the mount points. This mapping is done from the storage perspective, not the database perspective, so you must resolve all the symbolic links. You only need to map mount points that contain data files (i.e., the files in V\$DATAFILE). Example: The following is an example of the information you might enter with this key. <pre>data=<source_mntpoint_1>=> <clone_mntpoint_1>:<source_mntpoint_2>=><clone_mntpoint_2></pre>
bdump	The location of the background dump destination. This is the directory where Oracle generates the corresponding dumps for the clone. Just specify the directory pathname. Example: The following is an example of the information you might enter with this key. <pre>bdump=><clone_bdump_destination></pre>
cdump	The location of the clone dump destination. This is the directory where Oracle generates the corresponding dumps for the clone. Just specify the directory pathname. Example: The following is an example of the information you might enter with this key. <pre>cdump=><clone_cdump_destination></pre>
udump	The location of the user dump destination. This is the directory where Oracle generates the corresponding dumps for the clone. Just specify the directory pathname. Example: The following is an example of the information you might enter with this key. <pre>udump=><clone_udump_destination></pre>

Key	Value
archlog	A colon-delimited mapping of directories. This mapping is done from the database perspective, not the storage perspective, so you should not resolve the symbolic links. You must specify the directory mapping at a level that is deeper than the mount point level. You must specify a mapping for each archive log destination. Example: The following is an example of the information you might enter with this key. <pre>archlog=<source_archlog_dest1>=> <clone_archlog_dest1>:<source_archlog_dest2>=><clone_archlog_dest2> ...</pre>
redolog	A colon-delimited mapping of directories. This mapping is done from the database perspective, not the storage perspective, so you do not need to resolve the symbolic links. You can specify the directory mapping at a level that is deeper than the mount point level. You must specify a mapping for each redolog destination. Note_____ Make sure that there are no raw devices in the redo log groups. SnapManager's clone operation fails if the redolog group uses a raw device. _____ Example: The following is an example of the information you might enter with this key. <pre>redolog=<source_redolog_dest1>=> <clone_redolog_dest1>:<source_redolog_dest2>=><clone_redolog_dest2> ...</pre>
control	A colon-delimited list of absolute paths indicating the filenames that specify where SnapManager should generate the control files for the cloned database. Example: The following is an example of the information you might enter with this key. <pre>control=<clone_controlfile_1>:<clone_controlfile_2></pre>

Sample clone map file

The following is a sample clone map file.

```
data=/mnt/oracle/data=>/mnt/oracle/data_clone
bdump=/mnt/oracle/misc_clone/bdump
cdump=/mnt/oracle/misc_clone/cdump
udump=/mnt/oracle/misc_clone/udump
archlog=/mnt/oracle/misc/archlg=>/mnt/oracle/misc_clone/archlg:/mnt/oracle/misc/archlg
2=>/mnt/oracle/misc_clone/archlg2:/mnt/oracle/misc/archlg3=>/mnt/oracle/misc_clone/archlg3
redolog=/mnt/oracle/misc/redolg=>/mnt/oracle/misc_clone/redolg
control=/mnt/oracle/misc_clone/control_clone01.ctl:/mnt/oracle/misc_clone/control_clone02.ctl
```

Cloning a database

General steps for cloning a database from the CLI or GUI

Both the CLI command and the GUI wizard require similar information if you want to clone a database. This section takes you through the basic steps. For a syntax example, see [“Cloning a backup database from the CLI”](#) on page 121. For steps on using the GUI, see [“Cloning a backup database from the GUI”](#) on page 121.

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

1. Specify the name of the backup. This can be either:
 - ❖ The label you supplied when you created the backup
 - ❖ The 32-character HEX string GUID that SnapManager generated when you created the backup. (For more information about the GUID, see [“General notes about the commands”](#) on page 128.)
2. Supply a SID for the new, cloned database.
 - ❖ If you are cloning a RAC database and you want the clone to be a RAC database, enter the database name of the new RAC database.
 - ❖ If you want the clone to be a non-RAC database, enter the SID for the database.

Note

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

3. Specify the name of the profile from which the offline backup was created.
4. Specify the path to the file containing the clone mapping information. This can be an absolute or relative path. You must set up the clone map file before you start the clone operation.
5. (Optional) Specify how you want SnapManager to handle messages when executing the command from the CLI. For more information on these options, see [“Common options for handling messages shown on the console”](#) on page 130.

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

Note

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

Cloning a backup database from the CLI

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

Note

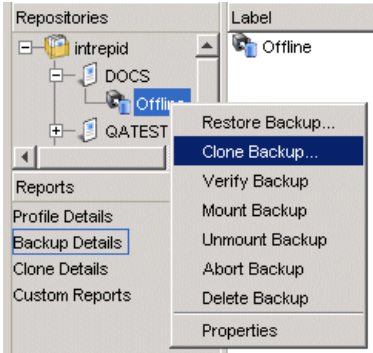
Make sure that the backup you are cloning is a full backup that was created while the database was offline.

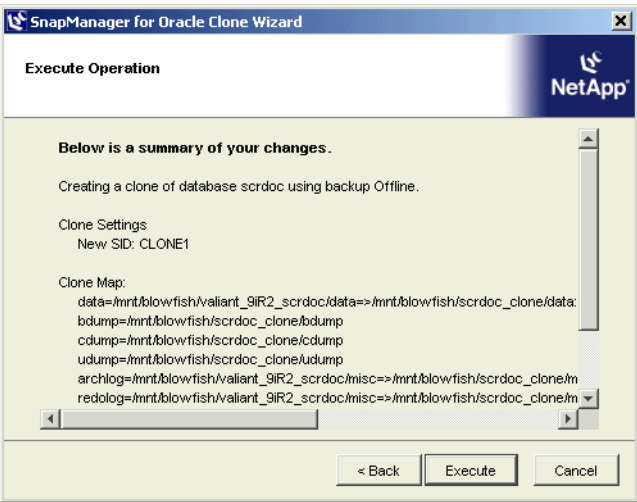
Step	Action
1	Create a clone map file. For instructions on how to do this, see “Creating a clone map file” on page 116.
2	Enter the <code>smo clone</code> command. It has the following general format: <pre>smo clone {-id <i>guid</i> -label <i>label_name</i>} -newsid <i>new_sid</i> -profile <i>profile_name</i> -clonemap <i>full_path_to_mapfile</i> [-quiet -verbose]</pre> For details on the options and arguments used in this command, see “The smo clone command” on page 137. Example: The following example creates a clone from a backup labeled Test1, which is included in the profile Alltests. The clone uses the map file drop1.clonemap, which is located on the host. <pre>smo clone -label Test1 -newsid EXC79D -profile Alltests -clonemap ./maps/test1.clonemap</pre>

Cloning a backup database from the GUI

To use the GUI's SnapManager for Oracle Clone Wizard to clone a backup of a database instance, complete the following steps.

Note—————
Make sure that the backup you are cloning is a full backup that was created while the database was offline.

Step	Action
1	Create a clone map file. For instructions on how to do this, see “ Creating a clone map file ” on page 116.
2	If the SnapManager GUI is not running, start it. For information on doing this, see “Starting the GUI” on page XX.
3	From the Navigation pane in the GUI, select the backup to clone. Select repository > profile to display the backups for that profile.
4	Select Clone Backup. You can select this operation from the menu or from the Tasks pane. The SnapManager for Oracle Clone Wizard starts. Note————— This example uses a backup from the profile DOCS in the repository intrepid. A screenshot of the SnapManager for Oracle GUI. The 'Repositories' pane on the left shows a tree structure with 'intrepid' as the root, containing 'DOCS' and 'QATEST' profiles. The 'DOCS' profile is selected. A context menu is open over the 'DOCS' profile, listing options: 'Restore Backup...', 'Clone Backup...' (highlighted), 'Verify Backup', 'Mount Backup', 'Unmount Backup', 'Abort Backup', 'Delete Backup', and 'Properties'. The 'Label' pane on the right shows 'Offline'.

Step	Action
7	At the Execute Operation screen, verify the information you've supplied. If you need to correct anything, click Back until you reach the screen containing the information you want to correct. If the information is correct and you want to create this clone, click Execute. 
8	The Completing the SnapManager for Oracle Clone Wizard screen appears. If you successfully created the clone, click Finish. If the clone creation failed, you can click Back to return to screens and correct the information.

Using cloning with RAC databases

Options for cloning RAC databases

When you have a RAC database, you can clone it to be either a RAC database or a non-RAC database. The cloned database is identical to the original database except that it is started as a normal database. SnapManager sets the Oracle parameter “cluster.database” to false.

Using a RAC database as a non-RAC database

You can use 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 -label rac_full_backup -profile rac_profile
  -newsid clone1 -clonemap rac_nfs.clonemap -verbose
$ export ORACLE_SID=clone1
$ sqlplus
...
```

Cloning a RAC database

If you want to clone a RAC database and then start a one node RAC instance, complete the following steps.

Step	Action
1	Use the database name of the new RAC database instance as the value for the -newsid option in the SnapManager smo clone command. Note _____ Do not use the Oracle SID as the name of the new RAC database instance.
2	When the cloning operation completes, rename the fileinit<clone_SID>.ora to init<clone_local_instance_SID>.ora. Then edit that file to set the RAC database instance information.
3	Register the cloned RAC database with srvctl.

Example of setting up a cloned database

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

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

About this chapter

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

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“Collecting information needed by SnapManager Commands”](#) on page 128
- ◆ [“Common options for handling messages shown on the console”](#) on page 130
- ◆ [“The smo backup command”](#) on page 131
- ◆ [“The smo check command”](#) on page 136
- ◆ [“The smo clone command”](#) on page 137
- ◆ [“The smo dump command”](#) on page 139
- ◆ [“The smo manage abort command”](#) on page 142
- ◆ [“The smo manage list command”](#) on page 144
- ◆ [“The smo manage mount command”](#) on page 145
- ◆ [“The smo manage remove backup command”](#) on page 147
- ◆ [“The smo manage unmount command”](#) on page 150
- ◆ [“The smo pfile command”](#) on page 152
- ◆ [“The smo profile create command”](#) on page 154
- ◆ [“The smo profile delete command”](#) on page 159
- ◆ [“The smo profile list command”](#) on page 161
- ◆ [“The smo profile show command”](#) on page 162
- ◆ [“The smo profile update command”](#) on page 164
- ◆ [“The smo restore and recovery command”](#) on page 169
- ◆ [“The smo verify command”](#) on page 173
- ◆ [“The smo version command”](#) on page 175

Collecting information needed by SnapManager Commands

Collecting information needed by commands

This chapter provides details that you can use as you execute SnapManager. For each command, it supplies the following:

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

The sections provide a quick overview of using these commands. For more information on the commands, see the chapter “[Backing Up and Restoring Databases](#)” on page 81.

General notes about the commands

The following are general notes about the commands:

- ◆ NetApp recommends that, when you create a backup, you supply a name for the backup using the `-label` option. This provides you with an easy to use name for 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 manage list` command.
- ◆ If the shell on your host limits the number of characters that can appear on a command line, you can use the `smo pfile` command. This command lets you put a SnapManager command in a text file and execute it that way. For more information, see “[The smo pfile command](#)” on page 152.
- ◆ If you are working online, you can get information about individual commands by issuing the `smo help` command. This command has the following format:

```
# smo help [backup|restore|clone|verify|manage|profile|check|dump]
```

Note

If you enter `smo help` with the `manage` option, you can get help on the `abort`, `list`, `remove`, `mount`, and `unmount` portions of the `smo manage` command.

Conventions for entering names and command options

Command options are case-sensitive

SnapManager command options are entered as 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.

The only exception to this is the profile name. Profile names are not case sensitive.

Avoid spaces in names

In general, SnapManager does not accept spaces in

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

If you are executing SnapManager from the GUI, it accepts profile names with spaces in them. The CLI, however, does not. For example, if you attempt to delete a profile from CLI and the profile name contains a space, the operation fails. You must use the GUI if you want to delete a profile that contains spaces in its name.

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

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

Limit the size of comments

SnapManager restricts comments to 255 characters.

If you supply a comment that is more than 255 characters and attempt to perform a backup operation, the operation fails without displaying a message.

Common options for handling messages shown on the console

Command options affecting the display of messages

For each command, SnapManager lets you specify where you want it 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 you specify using a command option, 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 command

Using the smo backup command

The `smo backup` command lets you create backups of databases on one or more storage system volumes. When the `smo backup` command completes, it returns a Globally Unique Identifier (GUID) for the backup. The GUID is a 32-character HEX string generated by the system. (For more information about the GUID, see [“General notes about the commands”](#) on page 128.)

Usage format for smo backup:

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

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

Note

Before you perform this command, you must create a database profile using the `smo profile` command. For information on how to do this, see [“The smo profile create command”](#) on page 154.

Requirement	Argument
Specify the state the database must be in when you perform the backup. ◆ <code>-auto</code>. SnapManager checks the state of the backup and determines which kind of backup to perform: ❖ If the archive log mode and database online mode are enabled, it performs an online backup of the database. ❖ If the database is in a mounted state, it performs an offline (cold) backup. ❖ If the <code>-force</code> option is specified and none of the above are true, it does an offline backup ◆ <code>-online</code>. SnapManager performs the backup while the database is up (i.e., a hot backup). ◆ <code>-offline</code>. SnapManager performs the backup while the database it is shut down (i.e., a cold backup). In this case, the database is mounted, but not open. If the database is up when you include this option, the backup fails. Note_____ If you specify the <code>-online</code> option when the database is shut down or the <code>-offline</code> option when the database is up, the backup fails. To prevent this failure, include the <code>-force</code> option on the command line to have SnapManager attempt to switch backup to the correct state. _____	
◆ <code>-online</code>	~
◆ <code>-offline</code>	~
◆ <code>-auto</code>	~
Specify whether you want SnapManager to back up the entire database or only part of it: ◆ When you specify a full backup, SnapManager makes a copy of the entire database. This includes all the data, log and control files. ◆ When you specify a partial backup, SnapManager copies only the portion of the database you specify and the log and control files. You can specify either a list of tablespace names (<code>-tablespaces</code>) or a list of file names (<code>-files</code>). Use commas to separate the names in the list. Do not include any spaces in these lists. If you specify filenames and the database is up, SnapManager ensures that the tablespace containing the files is in hot backup mode.	
◆ <code>-full</code>	~
◆ <code>-files list</code>	<i>comma-separated list of file names (no spaces)</i>
◆ <code>-tablespaces list</code>	<i>comma-separated list of tablespace names (no spaces)</i>

Requirement	Argument
Provide information about the database by entering the name of the profile for the database you want to back up. Note_____ You must have set up this information previously using the <code>smo profile</code> command. See the “The smo profile create command” on page 154. _____	
◆ <code>-profile name</code>	<i>name of the profile</i>
(Optional) Specify whether you want to SnapManager to verify that the files in the database were not corrupt. If you include the <code>-verify</code> option, SnapManager invokes the Oracle utility <code>dbv</code>. Note_____ If you include this option, the backup operation does not complete until the verify operation completes. _____ For more information about the <code>-verify</code> option, see the “Example of smo verify command line” on page 174.	
◆ <code>-verify</code>	~

Requirement	Argument
(Optional) Specify information about the backup. SnapManager lets you enter the following information: ◆ <code>-label</code>. Enter a name for the backup. This name must be unique for that profile. This option lets you supply an easy-to-recognize name for the backup. The name you supply can contain alphanumeric characters and the following special characters: underscore (<code>_</code>) and hyphen (<code>-</code>). It cannot start with a hyphen. If you do not supply a label, SnapManager creates a default label that has the format <code><scope>_<type>_<date></code> where ❖ <code>scope</code> is either F to indicate a full backup or P to indicate a partial backup. ❖ <code>type</code> is either C to indicate an offline (cold) backup or H to indicate an online (hot) backup. ❖ <code>date</code> is the year, month, day, and time of the backup. By default, SnapManager uses a 24-hour clock. For example, if you performed a full backup with the database offline on Jan. 16, 2006, at 5:45:16 p.m. Eastern standard time, SnapManager would create the label <code>F_C_20060116174516EST</code>. Whether you specify a label or not, SnapManager generates a GUID for the backup. (See “General notes about the commands” on page 128.) ◆ <code>-comment</code>. Supply a comment to describe this backup. This comment can be any single-quoted string. For example, you might enter ‘QA database’ so that you can quickly identify the contents of the backup. Note _____ Some shells strip off quote marks. If that is true for your shell, you must escape the quote with a backslash (<code>\</code>). For example, you might need to enter: <code>\ ' this is a comment \ '</code> to keep your shell from stripping off the quote marks. _____	
◆ <code>-label</code> <i>name</i>	<i>unique name you want to assign to the backup</i>
◆ <code>-comment</code> <i>comment</i>	<i>string to describe the backup</i>
(Optional) Decide if you want SnapManager to force a state change if the database is not in the correct state. For example, if you use this option, SnapManager might change the state of the database from online to offline, based on the type of backup you specify and the state that the database is in.	
◆ <code>-force</code>	<code>~</code>

Requirement	Argument
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo backup command line

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

```
# smo backup -online -tablespaces take1,users,hr
    -profile EXC79 -verify -label EXC79-Prod-Full
```

The smo check command

Using the smo check command

The `smo check` command lets you confirm that all the components of the environment required to run SnapManager are set up correctly.

Usage format for smo check:

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

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

Requirement	Argument
Enter the name of profile you created previously. SnapManager checks the environment for this profile. This profile defines all the credentials that SnapManager requires when working with a database. As part of the verification process, the <code>smo check</code> command confirms that the database included in this profile is accessible.	
◆ <code>-profile profile_name</code>	<i>name of an existing profile</i>
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo check command line

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

```
# smo check -profile testers
```

The smo clone command

Using the smo clone command

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

Usage format for smo clone:

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

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

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

Requirement	Argument
Specify the pathname to the file containing the mapping for the clone. This can be a relative or absolute pathname.	
◆ <code>-clonemap filename</code>	<i>path to the clone-mapping file</i>
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo clone command line

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

```
# smo clone -label EXC79-Prod-Full -newsid EXC79D -profile EXC79
```

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

The smo dump command

Using the smo dump command

The `smo dump` command lets you create a dump file containing information about your environment. If you specify a specific backup, SnapManager includes extra diagnostic information in the file. This file can be helpful if you need to troubleshoot a problem with the operation.

Usage format for smo dump:

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

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

Requirement	Argument
(Optional) Specify the name of the backup. This can be either of the following:	
◆ The default GUID that SMO generated when you created the backup. This is a 32-character HEX string. ◆ The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup</code> command line.	
◆ <code>-id <i>guid</i></code>	<i>GUID generated when you created the backup</i>
◆ <code>-label <i>label_name</i></code>	<i>name you supplied when you created the backup</i>
Specify the name of the profile associated with this backup.	
◆ <code>-profile <i>profile_name</i></code>	<i>name of the profile</i>
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130.	
Note _____	
Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file.	

◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo dump command line

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

```
# smo dump -label EXC79-Prod-Full -profile EXC79
```

The smo help command

Using the smo help command

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

Usage format for smo profile help:

```
smo help [command_name]
```

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

Requirement	Argument
(Optional) Include the command about which you want information. If you omit the this option, SnapManager displays a list of commands.	
◆ <i>command</i>	<i>SnapManager command</i>

Example of smo help command line

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

```
# smo help backup
```

The smo manage abort command

Using the smo manage abort command

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

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

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

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

Usage format for smo manage abort:

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

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

Requirement	Argument
Specify the backup that the operation you want to abort is using. You can identify the backup by either of the following:	
◆ The GUID that SMO generated when you created the backup. (For more information about GUIDs, see “General notes about the commands” on page 128.) ◆ The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup</code> command line.	
◆ <code>-id guid</code>	<i>GUID generated when you created the backup</i>
◆ <code>-label label_name</code>	<i>name you supplied when you created the backup</i>
Specify the name of the profile associated with this backup.	

Requirement	Argument
◆ <code>-profile profile_name</code>	<i>name of the profile</i>
(Optional) Decide if you want SnapManager to force the operation to abort. If you include this option, SnapManager aborts the operation without asking additional questions.	
◆ <code>-force</code>	~
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Example of smo manage abort command line

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

```
# smo manage abort -label EXC79-Prod-Full -profile EXC79
```

The smo manage list command

Displaying backup information

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

Usage format for smo manage list:

```
smo manage list -profile profile_name
```

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

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

Example of smo manage list command line

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

```
# smo manage list -profile EXC79
```

The smo manage mount command

Using the smo manage mount command

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

Note SnapManager automatically handles mounting backups if you use the smo restore command to restore the backup.

Usage format for smo manage mount:

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

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

Requirement	Argument
Specify the name of the backup that you want mounted. This can be either of the following:	
◆ The default GUID that SMO generated when you created the backup. This is a 32-character HEX string. ◆ The name you specified when the backup was created. You have the option of entering a name as the value for -label when you set up the smo backup command line.	
◆ -id guid	GUID generated when you created the backup
◆ -label label_name	name you supplied when you created the backup
Specify the name of the profile associated with the backup.	
◆ -profile profile_name	name of the profile

Requirement	Argument
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

**Example of smo
manage mount
command line**

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

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

The smo manage remove backup command

Using the smo manage remove backup command

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

You can use this command to remove backups that are not affected by the `-retain` option you set when you created or modified the profile (see [“The smo profile create command”](#) on page 154 and [“The smo profile update command”](#) on page 164.)

You can use this command to tell SnapManager to remove backups used as clones and failed backups.

Usage format for smo manage remove backup:

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

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

Requirement	Argument
Decide whether you want to delete a specific backup or limit the number of backups you have to a specific number. To delete a specific backup enter one of the following: ◆ <code>-id <i>guid</i></code>. The GUID that SMO generated when you created the backup. (For more information about GUIDs, see “General notes about the commands” on page 128.) ◆ <code>-label <i>label_name</i></code>. The name you specified when the backup was created. You have the option of entering a name as the value for <code>-label</code> when you set up the <code>smo backup</code> command line. To delete successful backups until you have only a specific number remaining, enter the following: ◆ <code>-retain <i>number</i></code>. This option tells SnapManager how many backups to keep. It keeps the number of backups you specify and removes all other backups. It removes the oldest backups first. For example, if you had five backups labeled May, June, July, August, and September and you put <code>-retain 3</code> on the command line, SnapManager would remove the backups May and June. If you specify 3 backups to be retained and you only have 2 backups, SnapManager does not remove any backups. To delete all the backups, set this value to 0.	
◆ <code>-id <i>guid</i></code>	<i>GUID generated when you created the backup</i>
◆ <code>-label <i>label_name</i></code>	<i>name you supplied when you created the backup</i>
◆ <code>-retain <i>number</i></code>	<i>number of backups to keep; if you enter 0, SnapManager removes all backups</i>
Specify the name of the profile associated with the backup you want to remove.	
◆ <code>-profile <i>profile_name</i></code>	<i>The name of the profile from which the backup was made</i>
(Optional) Decide if you want SnapManager to force the removal of the backup. When you include this option, SnapManager removes the backup even if there are problems freeing the resources associated with the backup. For example, if the backup was cataloged with RMAN, but the RMAN database no longer exists, including <code>-force</code> on the command line lets SnapManager delete the backup even though it cannot connect with RMAN.	
◆ <code>-force</code>	<i>~</i>

Requirement	Argument
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smc manage remove command line

The following are sample command lines that use the `smc manage remove` command.

```
# smc manage remove backup -retain 14 -profile EXC79
# smc manage remove backup -label EXC79-Prod-Full
  -profile EXC79
```

The smo manage unmount command

Using the smo manage unmount command

The `smo manage unmount` command lets you unmount a backup that you want to use in a recover operation. You should only need to use this command if you are using an external tool such as RMAN. You can use this command to free up resources on the host when you no longer need the backup.

Note SnapManager automatically handles unmounting backups if you use the `smo manage restore` command.

Usage format for smo unmount:

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

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

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

Requirement	Argument
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Examples of smo unmount command line

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

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

The smo pfile command

Using the smo pfile command

The `smo pfile` command lets you put commands in individual text files and have SnapManager execute the command that way.

Use this command if the shell on your host limits the number of characters that can appear on a command line. Enter the command line and its options exactly as you would if you were able to enter on the command line.

You can only put one command in each file.

Usage format for smo pfile:

`smo pfile=file_name`

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

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

Example of smo pfile command line

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

Suppose you have a text file called `commands.txt` in `/tmp` that has the following `smo profile create` command in it:

```
profile create TESTER -repository DB1 -host macbeth
-username bali -password high -port 1521 -database +ASM
-sid +ASM -username sys -password pacific -port 1521
-host florida -hostagent -username admin -password netapp
-OSAccount oracle -asm +ASM -username sys
-password aloha -port 1521 -host florida -rman -username rman
-password rmanuser -tnsname DB1
```

Now you can create the profile by entering the `sno pfile` with that file.

```
# sno pfile /tmp/commands.txt
```

The smo profile create command

Using the smo profile create command

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

Usage format for smo profile create:

```
smo profile create profile_name
    {-repository name -username username -password password
    -port port_number -host host [-sid sid]}
    {-database name -username username -password password -host host
    -port port_number [-sid sid]}
    [-rman [-controlfile | -username username -password password
    -tnsname tnsname] ]
    {-hostagent -username username -password password}
    {-OSAccount account_name}
    [-asm name -port port_number -username username
    -password password [-sid sid] ]
    [-retain number]
    [-quiet | -verbose]
```

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

Note_____

To update a profile, you use the command smo profile update instead of smo profile create.

Requirement	Argument
Specify the name you want to use for this profile. This name can be up to 30 characters long and must be unique for that host.	
◆ <i>profile_name</i>	<i>name of the profile</i>

Requirement	Argument
Enter the name of the repository that will contain this profile. Use either the global name or the SID. Keep in mind that this repository must reside in a different database from the database associated with this profile. This is because the repository contains both the credentials for the database and the names of any snapshots taken of the database during a backup. It keeps this information in an encrypted file. You must also specify the connection information for the repository. This is the information that SnapManager needs in order to connect to the Oracle database. You must supply: ◆ The database host. This can be either the host name or the host IP address. ◆ The username and password for connecting to the Oracle database containing the repository. ◆ The port number to use to connect.	
◆ <code>-repository name</code>	<i>name of the repository that will contain the profile</i>
❖ <code>-host host</code>	<i>name or IP address of the database host</i>
❖ <code>-username username</code>	<i>username for connecting to the database where the repository resides</i>
❖ <code>-password password</code>	<i>password for connecting to the database where the repository resides</i>
❖ <code>-port port_number</code>	<i>port number for connecting to the database</i>
You must specify the database to be operated on and the information SnapManager needs in order to connect to the database: ◆ The name of the database you want in this profile ◆ The database host (either the host name or its IP address). ◆ The username and password for connecting to the Oracle database. ◆ The port number to use to connect. ◆ In some cases, you must also include the SID. By default, SnapManager uses the database name as the SID. If the SID is different from the database name, you must specify it with the <code>-sid</code> argument. For instance, if you are using RAC, you must use this argument to specify the SID of the RAC instance on the RAC node from which SnapManager is executed.	
◆ <code>-database name</code>	<i>name of the database being profiled</i>
❖ <code>-host host</code>	<i>name of the database host</i>
❖ <code>-username username</code>	<i>username for connecting to the database</i>
❖ <code>-password password</code>	<i>password for connecting to the database</i>

Requirement	Argument
❖ <code>-port port_number</code>	<i>port number for connecting to the database</i>
❖ <code>-sid sid</code>	<i>SID for connecting to the database</i>
(Optional) If you want to catalog the SnapManager backups with RMAN, you must include the RMAN connection information in the profile also. To use RMAN, you must supply: ◆ The option <code>-rman</code>. ◆ (Optional) The <code>-controlfile</code> option if you want to use the control files instead of the RMAN catalog. ◆ If you do not use the <code>-controlfile</code> option, you must provide: ❖ The username and password for connecting to the RMAN catalog. ❖ The tnsname connection name (this is defined in the tnsname.ora file).	
◆ <code>-rman</code>	~
❖ <code>-controlfile</code>	~
❖ <code>-username username</code>	<i>username for connecting to the RMAN catalog</i>
❖ <code>-password password</code>	<i>password for connecting to the RMAN catalog</i>
❖ <code>-tnsname tnsname</code>	<i>connection name in the tnsname file</i>
You must specify the host agent and the username and password required to connect to it. Note _____ SnapManager requires that the NetApp Host Agent be installed. _____	
◆ <code>-hostagent</code>	~
❖ <code>-username username</code>	<i>username for connecting to the host agent</i>
❖ <code>-password password</code>	<i>password for connecting to the host agent</i>
You must specify the name of the Oracle database user account. This is the account that SnapManager uses to perform the Oracle operations such as startup and shutdown. It is typically the user who owns the Oracle software on the host; for example, oracle.	
◆ <code>-osaccount owner_name</code>	<i>name of the Oracle database owner account</i>

Requirement	Argument
(Optional) If you use the Oracle Automatic Storage Management (ASM) instance, you must specify the information SnapManager needs in order to connect to it: ◆ The username and password for connecting to the Oracle database. ◆ The port number to use to connect. ◆ In some cases, you must also include the SID. By default, SnapManager uses the database name as the SID. If the SID is different from the database name, you must specify it with the <code>-sid</code> argument. For instance, if you are using RAC, you must use this argument to specify the SID of the RAC instance on the RAC node from which SnapManager is executed.	
◆ <code>-asm name</code>	<i>name of the ASM instance</i>
❖ <code>-username username</code>	<i>username for connecting to ASM</i>
❖ <code>-password password</code>	<i>password for connecting to ASM</i>
❖ <code>-port port_number</code>	<i>port number for connecting to ASM</i>
❖ <code>-sid sid</code>	<i>SID for connecting to the database</i>
(Optional) This option applies only to successful backups that were not used to clone a database. It does not affect unsuccessful backups, which SnapManager keeps, or backups used to clone a database. Specify the number of successful backups you want to SnapManager to retain. Enter an integer. If you do not supply a number here, SnapManager replaces the previous backup each time it creates a new backup. Note_____ You can specify any integer you like here; SnapManager does not have a limit. If you keep every backup, you may eventually reach the snapshot limit or run out of space. _____ When you include the retain option, SnapManager keeps the most recent successful backups up to the number you specify. When SnapManager produces a successful backup that causes it to exceed this number, it deletes the oldest successful backup, unless that backup was used to clone a database. SnapManager does not delete backups used to create clones or unsuccessful backups. You must execute the <code>smo manage remove</code> operation to remove those backups. Note_____ The unsuccessful backups do not affect the storage. Only the successful backups require space on the storage. _____	
◆ <code>-retain number</code>	<i>number of backups that are kept</i>

Requirement	Argument
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the -quiet option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ -quiet	~
◆ -verbose	~

Example of smo profile create command line

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

```
# smo profile create TESTER -repository DB1 -host macbeth
  -username bali -password high -port 1521 -database +ASM
  -sid +ASM -username sys -password pacific -port 1521
  -host florida -hostagent -username admin -password netapp
  -OSAccount oracle -asm +ASM -username sys
  -password aloha -port 1521 -host florida -rman -username rman
  -password rmanuser -tnsname DB1
```

The smo profile delete command

Using the smo profile delete command

The smo profile delete command lets you delete a profile of a database.

Usage format for smo profile delete:

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

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

Requirement	Argument
Specify the name you of the profile you want to delete.	
◆ <i>profile_name</i>	<i>name of an existing profile</i>
(Optional) Decide if you want SnapManager to attempt to force the deletion. If the profile has backups associated with it, the delete operation fails unless you include this option. When you include the <code>-force</code> option, SnapManager attempts to delete all backups for that profile. Even with the <code>-force</code> option on the command line, the profile delete operation could fail if the database is not in the correct state. This is because SnapManager does not change the database state in order to delete backups during a profile delete. If the SnapManager cannot delete the backups in the current database state, you must either: ◆ Manually remove the backups using the <code>smo manage remove</code> command. For example, if the database is shutdown and the backups are cataloged with RMAN, you must first manually remove the backups before you can perform a successful profile delete operation. ◆ Put the database in the correct state before you perform the profile delete operation.	
◆ <code>-force</code>	~
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~

Requirement	Argument
◆ -verbose	~

Example of smo profile delete command line

The following is a sample command line that uses the `smo profile delete` command.

```
# smo profile delete TESTER
```

The smo profile list command

Using the smo profile list command

The smo profile list command lets you display a list of the current profiles. If you include the `-verbose` option with this command, it displays the contents of each profile.

Usage format for smo profile list:

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

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

Requirement	Argument
(Optional) Include the <code>-verbose</code> option if you want SnapManager to display the build date and the contents of each profile. If you omit the this option, SnapManager only displays the profile descriptor.	
◆ <code>-verbose</code>	~

Example of smo profile list command line

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

```
# smo profile list -verbose
```

The smo profile show command

Using the smo profile show command

The smo profile show command lets you display information about a profile. If you include the `-verbose` option with this command, it displays the contents of the profile you specify.

Usage format for smo profile show:

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

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

Requirement	Argument
Specify the name of the profile that you want to display information on.	
◆ <i>profile_name</i>	<i>name of an existing profile</i>
(Optional) Include the <code>-verbose</code> option if you want SnapManager to display the profile contents. Without this option, SnapManager only displays the profile descriptor. If you omit the this option, SnapManager only displays the version number and the profile name.	
◆ <code>-verbose</code>	~

Example of smo profile show command line

The following is a sample command line that uses the `smo profile show` command.

```
# smo profile show qa_prof -verbose
# smo profile show qa_prof -verbose
SMO-13037 [INFO ]: Successfully completed operation: Profile
SMO-13048 [INFO ]: Operation Status: SUCCESS
SMO-13049 [INFO ]: Elapsed Time: 0:00:05.101
SMO-13990 [INFO ]: Snap Manager for Oracle version
0.0.633N20060117_0953.
SMO-13036 [INFO ]: Starting operation: Profile
-----
Repository descriptor: /u/smo/profiles/qa_prof.profile
URL: jdbc:oracle:thin:@//polo:1521/SMOREPO
Host: polo
```

```
Username: tester
Database: SMOREPO
Profile: qa_prof
Backups to retain: 5
Database name: db1
Database sid: db1
Host: polo
Port: 1521
User: sys
RMAN Database name: db1
RMAN Database sid: BTNFS
RMAN Host:
RMAN Port:
RMAN User:
Oracle user host: oracle
Host agent host name: polo
Host agent user: admin
SMO-13037 [INFO ]: Successfully completed operation: Profile
SMO-13048 [INFO ]: Operation Status: SUCCESS
SMO-13049 [INFO ]: Elapsed Time: 0:00:05.101
SMO-13026 [INFO ]: Operation succeeded.
```

The smo profile update command

Using the smo profile update command

The smo profile update command lets you modify the information in an existing profile of a database. The only differences between this command and the smo create command are that the profile must already exist and you use *update* instead of *create*.

You must use this command if any of the user name/password information associated with the database change. If you use the command to change your user name or password, make sure you first change the information on the host or database.

When you update a profile, you only need to include the block of command options that apply to the section you are changing. For example, if you want to update the repository information, you would only need to supply the profile name and the block of options that affect the repository.

Usage format for smo profile update:

```
smo profile update profile_name
  {-repository name -username username -password password
   -port port_number -host host [-sid sid]}
  {-database name -username username -password password -host host
   -port port_number [-sid sid]}
  [-rman [-controlfile | -username username -password password
   -tnsname tnsname] ]
  {-hostagent -username username -password password}
  {-OSAccount account_name}
  [-asm name -port port_number -username username
   -password password [-sid sid] ]
  [-retain number] [-remove {-asm name | rman} ]
  [-quiet | -verbose]
```

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

Note

To create a profile, you use the command `smo profile create` instead of `smo profile update`.

Requirement	Argument
Specify the name of the profile you want to update.	
◆ <i>profile_name</i>	<i>name of an existing profile</i>
Enter the name of the repository that containing the profile. You must also specify the connection information for the repository. This is the information that SnapManager needs in order to connect to the Oracle database. You must supply: ◆ The database host. This can be either the host name or the host IP address. ◆ The username and password for connecting to the Oracle database containing the repository. ◆ The port number to use to connect.	
◆ <i>-repository name</i>	<i>name of the repository that will contain the profile</i>
❖ <i>-host host</i>	<i>name or IP address of the database host</i>
❖ <i>-username username</i>	<i>username for connecting to the database where the repository resides</i>
❖ <i>-password password</i>	<i>password for connecting to the database where the repository resides</i>
❖ <i>-port port_number</i>	<i>port number for connecting to the database</i>
You must specify the database to be operated on and the information SnapManager needs in order to connect to the database: ◆ The name of the database you want in this profile ◆ The database host (either the host name or its IP address). ◆ The username and password for connecting to the Oracle database. ◆ The port number to use to connect. ◆ In some cases, you must also include the SID. By default, SnapManager uses the database name as the SID. If the SID is different from the database name, you must specify it with the <i>-sid</i> argument. For instance, if you are using RAC, you must use this argument to specify the SID of the RAC instance on the RAC node from which SnapManager is executed.	
◆ <i>-database name</i>	<i>name of the database being profiled</i>
❖ <i>-host host</i>	<i>name of the database host</i>
❖ <i>-username username</i>	<i>username for connecting to the database</i>
❖ <i>-password password</i>	<i>password for connecting to the database</i>

Requirement	Argument
❖ <code>-port port_number</code>	<i>port number for connecting to the database</i>
❖ <code>-sid sid</code>	<i>SID for connecting to the database</i>
(Optional) If you want to catalog the SnapManager backups with RMAN, you must include the RMAN connection information in the profile also. To use RMAN, you must supply: ◆ The option <code>-rman</code>. ◆ (Optional) The <code>-controlfile</code> option if you want to use the control files instead of the RMAN catalog. ◆ If you do not use the <code>-controlfile</code> option, you must provide: ❖ The username and password for connecting to the RMAN catalog. ❖ The tnsname connection name (this is defined in the tnsname.ora file)	
◆ <code>-rman</code>	~
❖ <code>-controlfile</code>	~
❖ <code>-username username</code>	<i>username for connecting to the RMAN catalog</i>
❖ <code>-password password</code>	<i>password for connecting to the RMAN catalog</i>
❖ <code>-tnsname tnsname</code>	<i>connection name in the tnsname file</i>
You must specify the host agent and the username and password required to connect to it. Typically, this username is admin. Note_____ SnapManager requires that the NetApp Host Agent be installed. _____	
◆ <code>-hostagent</code>	~
❖ <code>-username username</code>	<i>username for connecting to the host agent</i>
❖ <code>-password password</code>	<i>password for connecting to the host agent</i>
You must specify the name of the Oracle database user account. This is the account that SnapManager uses to perform the Oracle operations such as startup and shutdown. It is typically the user who owns the Oracle software on the host; for example, oracle.	
◆ <code>-osaccount owner_name</code>	<i>name of the Oracle database owner account</i>

Requirement	Argument
(Optional) If you want to use the ASM instance, you must specify the information SnapManager needs in order to connect to it: ◆ The username and password for connecting to the Oracle database. ◆ The port number to use to connect. ◆ In some cases, you must also include the SID. By default, SnapManager uses the database name as the SID. If the SID is different from the database name, you must specify it with the <code>-sid</code> argument. For instance, if you are using RAC, you must use this argument to specify the SID of the RAC instance on the RAC node from which SnapManager is executed.	
◆ <code>-asm name</code>	<i>name of the ASM instance</i>
❖ <code>-username username</code>	<i>username for connecting to ASM</i>
❖ <code>-password password</code>	<i>password for connecting to ASM</i>
❖ <code>-port port_number</code>	<i>port number for connecting to ASM</i>
❖ <code>-sid sid</code>	<i>SID for connecting to the database</i>
(Optional) The <code>-retain</code> option applies only to successful backups that were not used to clone a database. It does not affect unsuccessful backups, which SnapManager keeps, or backups used to clone a database. Specify the number of successful backups you want to SnapManager to retain. Enter an integer. If you do not supply a number here, SnapManager replaces the previous backup each time it creates a new backup. Note_____ You can specify any integer you like here; SnapManager does not have a limit. If you keep every backup, you may eventually reach the snapshot limit or run out of space. _____ When you include the retain option, SnapManager keeps the most recent successful backups up to the number you specify. When SnapManager produces a successful backup that causes it to exceed this number, it deletes the oldest successful backup, unless that backup was used to clone a database. SnapManager does not delete backups used to create clones or unsuccessful backups. You must execute the <code>sno delete</code> operation to remove those backups. Note_____ The unsuccessful backups do not affect the storage. Only the successful backups require space on the storage. _____	
◆ <code>-retain number</code>	<i>number of backups that are kept</i>

Requirement	Argument
(Optional) If you are using ASM or RMAN, you can use the <code>-remove</code> option to remove them from the profile. Specify either the name of the ASM instance or rman.	
◆ <code>- remove</code> ◆ <code>-asm name</code> ◆ <code>rman</code>	<i>specify either the name of the ASM instance or rman</i>
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Example of smo profile update command line

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

```
# smo profile update MYTEST -database mytestbed
    -username system -password pacific -host ocean -port 1521
```

The smo restore and recovery command

Using the smo restore command

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

Usage format for smo restore:

```
smo restore {-id guid | -label label_name}  
            -profile profile_name  
            {-complete | -files list | -tablespaces list | -controlfiles}  
            [-recover [-when {auto |scn | date } ] ]  
            [-force] [-quiet | -verbose]
```

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

Requirement	Argument
Specify the name of the backup that you want to restore. This can be either of the following: ◆ The default GUID that SMO generated when you created the backup. ◆ The name you specified when the backup was created. You have the option of entering a name as the value for -label when you set up the smo backup command line.	
◆ -id guid	<i>GUID generated when you created the backup</i>
◆ -label label_name	<i>name you supplied when you created the backup</i>
Specify the name of the profile associated with this backup.	
◆ -profile profile_name	<i>name of the profile</i>

Requirement	Argument
Specify whether you want SnapManager to restore the entire database or only part of it: ◆ When you specify the complete option, SnapManager restores everything in the backup. ◆ When you specify a partial backup, SnapManager restores only the portion of the backup you specify. You can specify either a list of tablespace names (<code>-tablespaces</code>) or a list of file names (<code>-files</code>). Use commas to separate the names in the list. Do not include spaces in the list. If you specify filenames and the database is up, SnapManager ensures that the tablespace containing the files is in offline. Note _____ If you are restoring data files, make sure you first restore the control files. Then repeat the operation and restore the data files. _____ ◆ When you specify the control files option, SnapManager restores only the control files.	
◆ <code>-complete</code>	~
◆ <code>-files list</code>	<i>a list of file names separated by commas (no spaces allowed)</i>
◆ <code>-tablespaces list</code>	<i>list of tablespace names separated by commas (no spaces allowed)</i>
◆ <code>-controlfiles</code>	~

Requirement	Argument
(Optional) If you want SnapManager to recover the files, include the <code>-recover</code> option. Then specify the point in time to which you want SnapManager to recover the database. SnapManager provides the following options based on whether you are performing a complete restore of the backup or a partial restore: ◆ Complete or partial restoration of the database. ❖ <code>auto</code>. SnapManager recovers the database to the last transaction or commit. Note _____ If you do not include <code>-when</code> with the <code>-recover</code> option, it defaults to an auto recovery. ◆ Complete restoration of the database. ❖ <code>date</code>. SnapManager recovers the database up to the date and time you specify. Use the format year:month:date:hour:minute:second (<code>yyyy-mm-dd:HH:MM:SS</code>). For example, you might enter 2006-01-25:19:06:22 to specify that the backup should be restored up to Nov. 11, 2005 at 7:06 p.m. 22 seconds. (Whether you use a 24-hour clock or not depends on how you've set up the options for your Oracle database.) Note _____ This is the time associated with the database. It is not necessarily the clock time on the host. ❖ <code>scn</code>. SnapManager rolls forward the Oracle SCN log files until it reaches the SCN number you specify. For example, you might enter 384641. SnapManager would recover the backup up to that SCN.	
◆ <code>-when auto</code> (complete or partial)	~
◆ <code>-when yyyy-mm-dd:HH:MM:SS</code> (complete restore only)	<i>enter the date and time up to which you want the recovery operation to continue</i>
◆ <code>-when scn</code> (complete restore only)	<i>integer representing the SCN number to which you want the recovery operation to continue</i>
(Optional) Decide if you want SnapManager to change the database state to a lower state than its current state. For RAC, you must include the <code>-force</code> option if SnapManager needs to change the state of any RAC instance to a lower state. By default, SnapManager can change the database state to a higher state during an operation. You do not need to enter this option for SnapManager to change the database to a higher state.	
◆ <code>-force</code>	~

Requirement	Argument
(Optional) Decide if you want messages from the CLI displayed on the console or only written to a log file. By default, SnapManager displays the error messages on the console and writes the messages to the log file. For details on these options, see “Common options for handling messages shown on the console” on page 130. Note_____ Even if you specify the <code>-quiet</code> option, SnapManager still displays error messages on the console in addition to writing them to the log file. _____	
◆ <code>-quiet</code>	~
◆ <code>-verbose</code>	~

Example of smo restore command line

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

```
# smo restore -complete -label EXC79-Prod-Full
  -profile EXC79 -recover -when auto
```

The smo verify command

Using the smo verify command

The `smo 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` command, it automatically executes the command after it completes the backup. In fact, the backup operation does not complete until the verify option finishes.

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

Usage format for smo verify:

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

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

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

Requirement	Argument
◆ -force	~

Example of smo verify command line

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

```
# smo verify -label EXC79-Prod-Full -profile EXC79
```

The smo version command

Using the smo version command

The smo version command lets you determine which version of SnapManager you are running on your local host. If you include the -verbose option with this command, it also displays the build date.

Usage format for smo version:

```
smo version [-verbose]
```

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

Requirement	Argument	
(Optional) Decide if you want messages from the CLI displayed on the console. If you include the <code>-verbose</code> option, SnapManager displays the build date and the contents of each profile. If you omit the this option, SnapManager only displays the version number and the profile name.		
◆ <code>-verbose</code>	<code>~</code>	

Example of smo version command line

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

```
# smo version -verbose
```


About this chapter

This chapter provides information about the troubleshooting tools and tips that are available to help you. These tools are useful for gathering information as well as solving problems.

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“Questions to ask when you encounter a problem”](#) on page 178
- ◆ [“Using the dump command to gather information”](#) on page 179
- ◆ [“General troubleshooting tips for using the GUI”](#) on page 182
- ◆ [“General troubleshooting tips for working with clones”](#) on page 183
- ◆ [“General troubleshooting tips for SnapManager”](#) on page 185
- ◆ [“Common error conditions”](#) on page 188

Questions to ask when you encounter a problem

Diagnosing problems

Sometimes a problem might occur while you are using SnapManager. If that happens, the following list of questions could help you resolve the issue:

- ◆ Are the target database and the listener running?

To check this, run the command:

```
lsnrctl status.
```

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

- ◆ Is the storage visible and writable?

Use the commands `df` and `sanlun lun show` to see if the storage is visible. If so, go to the directory and check the permissions on a new file in it to make sure that you can write to the storage system.

- ◆ Is the NetApp Host Agent configured and running?

To check the status, run the command:

```
/usr/sbin/ntap_agent status
```

To stop the NetApp Host Agent, run the command:

```
/usr/sbin/ntap_agent stop
```

To start the NetApp Host Agent, run the command:

```
/usr/sbin/ntap_agent start
```

- ◆ Can you run the `smo check` 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 check -profile profile_name [-quiet | -verbose]
```

For more information on this command, see “[The smo check command](#)” on page 136.

- ◆ 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. It writes multiple operations to the same log file.

By default, the log files are in `/opt/NetApp/smo/log` on a Linux host and `/opt/NTAPsmo/log` on a Solaris host.

Using the dump command to gather information

Getting information using the dump command

SnapManager can create a compressed log file containing information about the SnapManager and its environment. If you specify a specific backup, SnapManager includes extra diagnostic information in the file. The information in the file includes:

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

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

Locating the dump file

When SnapManager performs a dump operation, it creates a file with the name of either:

- ◆ `smo_dump_<id>.jar` where *<id>* is either the GUID or backup label. You must include either the `-id backup-name` or `-label label-name` option on the command line to get the name in this format.
- ◆ `smo_dump_results.jar` SnapManager uses this format of the name when you do not include either the GUID or backup label on the command line.

By default, SnapManager creates the dump file in the `/tmp` directory.

Information included in the dump file

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

Note

When you specify the dump operation, you have the option of supplying the name of the backup. SnapManager only reports storage system information if you provide either the GUID or the label for the backup.

The dump file contains the following information:

- ◆ Operating system version
- ◆ Environment variables

- ◆ Java version
- ◆ SnapManager version
- ◆ SnapManager preferences
- ◆ SnapManager messages
- ◆ log4j properties
- ◆ SnapDrive version
- ◆ NetApp Host Agent version
- ◆ Oracle version
- ◆ Storage system version for the storage system containing the backup (you only get this information if you include the backup name when you execute the dump operation)
- ◆ Oracle oratab file
- ◆ Oracle listener status
- ◆ Oracle network configuration files (listener.ora and tnsnames.ora)
- ◆ Profile descriptor
- ◆ smo.log
- ◆ ntap_agent.log
- ◆ Shared memory maximum
- ◆ Swap space information
- ◆ Memory information
- ◆ Kernel parameters
- ◆ FSTAB
- ◆ (SnapManager 1.1.1 only) SnapDrive for UNIX trace log
- ◆ (SnapManager 1.1.1 only) SnapDrive for UNIX audit log
- ◆ (SnapManager 1.1.1 only) SnapDrive for UNIX configuration information

General steps for creating a dump file from the CLI

This section takes you through the basic steps of creating a dump file. For a syntax example, see [“Creating a dump file from the CLI”](#) on page 181.

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

1. (Optional) Specify the name of the backup for which you want information. This name 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 how you want SnapManager to handle messages. By default, SnapManager displays the error and warning messages on the console. To change this 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 130.

Creating a dump file from the CLI

The `smo dump` command has the following general format:

```
smo dump [-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 dump command](#)” on page 139.

General troubleshooting tips for using the GUI

Make sure you have the correct version of the JRE

The SnapManager GUI requires that you have the Sun JRE 1.4.2 installed. You must install this product before you can run the GUI.

A copy of this version of the JRE is included on the SnapManager CD. It is also part of the software that is available for downloading from the SnapManager page on the NOW site.

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 Load Backup List option from the pop-up menu. SnapManager displays the Profile Authentication dialog box. Enter the database user name and password. Once you do this, SnapManager displays the backup list.

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

The verify operation succeeds, but its output is not available

Symptom: You run the verify operation on a backup and it succeeds. When you go to check the results, though, SnapManager says:

```
Backup has not been verified. No verification information to display.
```

Explanation: This sometimes happens when you are executing SnapManager from the GUI. It indicates that SnapManager has not updated its display after completing the command.

What to do: Stop and restart the GUI. This forces SnapManager to update its display.

General troubleshooting tips for working with clones

Avoid using a duplicate SID for the clone name

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

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

Clone attempt fails saying mount point 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 mount point. This means it is possible that an incomplete clone did not remove the mount point.

What to do: Run the `smo manage 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 -newsid racspcl -profile SMORMANDBASE -label CLONETP
```

You would then run the following `smo manage remove` command:

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

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

NID-00100 error occurs during clone operation

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

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

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

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

Clone operation fails when redo log group uses a raw device

Symptom: The clone operation fails when the redo log group uses a raw device.

Explanation: While SnapManager allows raw devices, it does not allow redo log groups on raw devices during clone operation.

What to do: Make sure your redo groups do not include any raw devices.

General troubleshooting tips for SnapManager

SnapManager fails to execute after installing NetApp Host Agent

Symptom: SnapManager does not execute. This problem occurred after you installed (or reinstalled) the NetApp Host Agent.

Explanation: When you install SnapManager, it places a library file in the `/opt/NTAPagent` directory. If you then uninstall and reinstall the NetApp Host Agent, it removes the contents of this directory, including the file SnapManager placed there.

What to do: Run the `smo_setup` script to replace the library file. This script is located in `/opt/NetApp/smo/bin/smo_setup` on Linux hosts and `/opt/NTAP/smo/bin/smo_setup` on Solaris hosts.

Backups registered in RMAN are not visible in the backup list

Symptom: The `list backup` 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 `list backup` command.

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

Recovering control files takes a long time

Symptom: You attempted to perform a controlfile 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.

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 database is stored on a SnapDrive for UNIX-managed file system on a LUN residing on a qtree.

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

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

Backup operation hangs indefinitely

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

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

What to do: If SnapManager appears to be hanging, use the `smo manage 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 allows 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 remove` 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.

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	17000-7999	Messages associated with the Virtual Database Interface. These messages deal with the interface from the product to the various Oracle versions.
VIRTUAL STORAGE INTERFACE	18000-8999	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.

Index

A

- abort operation
 - CLI command 142
 - CLI example 143
 - command parameters 142
 - executing from CLI 100
 - executing from GUI 100
 - general steps 99
 - graceful cleanup 98
 - incomplete backup 98
 - interrupting executing operation 98
- archlog
 - clone map file key 118
- ASM
 - backup fails with raw LUNs on qtree 185
 - Linux limitation 12
 - using with SnapManager 9
- authenticating
 - profile 69
 - repository 63
- auto recovery
 - defined 5
- Automatic Storage Management *See* ASM

B

- backup 30
 - cloned 30
 - cloning 114
 - defined 4
 - display profile information 161
 - displaying information about 101
 - displaying list 78
 - examples of retaining 30
 - labels 128
 - problem seeing in RMAN 185
 - removing 104
 - repository 19
 - retaining
 - retaining backups 30
 - sliding window 30
 - steps to display information 101

- unable to display in GUI 182
 - unsuccessful 30
- backup operation
 - algorithm for full backup 27
 - algorithm for partial backup 28
 - auto option 85
 - CLI command 131
 - CLI example 135
 - command parameters 131
 - executing from CLI 84
 - executing from GUI 84
 - fails without message 186
 - files included 21
 - full 26
 - hangs 186
 - incomplete 98
 - information required 82
 - offline backup 82
 - online backup 82
 - partial 26
 - types of backups 21
 - verifying 88
 - viewing files 21
- bdump
 - clone map file key 117

C

- case-sensitive
 - command options 129
- cdump
 - clone map file key 117
- check operation
 - CLI command 136
 - CLI example 136
 - command parameters 136
- checklists
 - installation tasks 36
- CLI
 - abort command 142
 - clone operation 137
 - dump command 139, 179

- executing abort operation 100
- executing backup operation 84
- executing clone operation 121
- executing list operation 102
- executing mount operation 108
- executing remove operation 105
- executing restore operation 92
- executing SnapManager 10
- executing unmount operation 108
- executing verify operation 88
- help command 141
- list command 144
- modifying profile 70
- mount vp,,smf 145
- pfile command 152
- profile create command 69, 154
- profile delete command 71, 159
- profile list command 161
- profile show command 162
- profile update command 164
- remove command 147
- restore command 169
- starting SnapManager 62
- unmount command 150
- verify command 173
- version command 175
- clone map file
 - archlog key 118
 - bdump key 117
 - cdump key 117
 - control key 118
 - creating 116
 - data key 117
 - example 119
 - key-value pairs 117
 - redolog key 118
 - udump key 117
- clone operation
 - CLI command 137
 - CLI example 138
 - command parameters 137
 - executing from CLI 121
 - executing from GUI 121
 - fails with NID-00100 183
 - general steps 120
 - problem with mount point 183
 - use unique SID 183
- cloning backups 114
 - clone map file 116
 - cloning RAC database 125
 - creating Oracle files 115
 - different SIDs 115
 - files created 115
 - RAC database 115
 - setting up RAC clone 126
- cloning backus
 - deleting 30
- Command Line Interface *See* CLI
- command options
 - case-sensitive 129
 - displaying messages 130
 - quiet 130
 - verbose 130
- commands
 - backup 131
 - check 136
 - clone 137
 - dump 139
 - excuting from GUI
 - 64
 - general information 128
 - help 141
 - manage abort 142
 - manage list 144
 - manage mount 145
 - manage remove 147
 - manage unmount 150
 - Oracle dbv 88
 - pfile 152
 - profile create 154
 - profile delete 159
 - profile list 161
 - profile show 162
 - profile update 164
 - recover option 169
 - restore 169
 - verify 173
 - version 175

Compatibility and Configuration Guide for

NetApp's FCP and iSCSI Products 13

control

clone map file key 118

controlfile

recovery slow 185

controlfiles

included in backup 21

conventions

formatting ix

keyboard x

credentials

encrypted 9

D

data

clone map file key 117

Data ONTAP Block Access Management Guide 14

database 7

recovering 90

restoring 90

databases

cloning 115

locked during operations 17

RAC 115

DBA

using SnapManager 34

deleting

cloned backup 30

unsuccessful backup 30

DES

encrypts credentials file 9

documentation

additional 13

Compatibility and Configuration Guide for
NetApp's FCP and iSCSI Products 13

Data ONTAP Block Access Management
Guide 14

FCP host attach kit 14

NOW Description Page 13

SnapDrive for UNIX Quick Start 14

SnapManager Compatibility Matrix 13

SnapManager Release Notes 13

System Configuration Guide 14

dump command

CLI command 139, 179

CLI example 140

executing 180

information producted 179

parameters 139

dump file

creating 181

locating 179

E

environment

verifying from GUI 79

environment variables

checking 79

error messages 130

code ranges 188

examples

abort command 143

backup command 135

check command 136

clone command 138

clone map file 119

dump command 140

help command 141

list command 144

mount command 146

pfile command 152

profile create command 158

profile delete command 160

profile list command 161

profile show command 162

profile update command 168

recover option 172

remove command 149

restore command 172

retaining backups 30

unmount command 151

verify create command 174

version command 175

executing SnapManager 10

F

FCP host attach kit documentation 14

file system

accepted by SnapManager 11

full 186

files

.pkg (Solaris) 45

.rpm (Linux) 42

log files 112

FlexClone

required 37

full backup

algorithm 27

compared to partial 26

G

Graphical User Interface *See* GUI

GUI

auto backup option 85

creating profile 72

deleting profile 78

displaying backup list 78

executing 10

executing abort operation 100

executing backup operation 84

executing clone operation 121

executing list operation 102

executing mount operation 109

executing restore operation 92

executing unmount operation 109

executing verify operation 89

Information and Reports window 66

installing 52

JRE version 182

main screen 65

modifying profile 77

navigating 16

Navigation pane 65

Reports pane 65

starting SnapManager 63

Tasks pane 66

troubleshooting 182

unable to display backups 182

using 64

verify operation fails 182

GUID

defined 4

view using list 128

H

help command

CLI example 141

command parameters 141

hosts

installation steps on Linux host 42

I

in-band events 33

Information and Reports window

GUI 66

informational messages 130

installing

GUI 52

loading the repository 49

steps for Linux host 42

upgrading SnapManager to a new version 53

J

JRE

required for GUI 37

L

label option

case sensitive 128

licenses

required 37

Linux host

.rpm file 42

example of installing 42, 54

example of uninstalling SnapManager 44

installation steps for SnapManager 42

requires ASMLib 12

uninstalling SnapManager 44

- list operation
 - CLI command 144
 - CLI example 144
 - command parameters 144
 - executing from CLI 102
 - executing from GUI 102
 - general steps 101

- lock
 - during operations 17

- log files
 - file system full 186
 - included in backup 21
 - location 112
 - rotating 112

M

- manage command
 - abort 142
 - list 144
 - mount 145
 - remove 147
 - unmount 150

- matrix
 - SnapManager compatibility 10

- mount operation
 - CLI command 145
 - CLI example 146
 - command parameters 145
 - executing from CLI 108
 - executing from GUI 109
 - general steps 107

- mount point
 - problem with clone 183

N

- navigating
 - GUI 16

- Navigation pane
 - GUI 65

- NetApp Host Agent
 - problem after installing 185
 - required 10, 37
 - restarting 50

- NID-00100 error 183
- NOW site
 - downloading software 41
 - SnapManager matrix 10

O

- OCFS
 - unsupported 12
- offline
 - defined 5
- offline backup 82
- online
 - defined 5
- online backup 82
- operational states
 - failed 32
 - in progress 32
 - in-band (recoverable) 33
 - incomplete 32
 - out-of-band (unrecoverable) 33
 - SnapManager 32
 - succeeded 32
- out-of-band
 - events 33

P

- partial backup
 - algorithm 28
 - compared to full 26
- permissions
 - setting 43, 47
- persistent state
 - maintained by SnapManager 98
- pfile command
 - CLI command 152
 - CLI example 152
 - command parameters 152
- PIT recovery
 - defined 5
- pkg file
 - Solaris installation 45
- Point-in-time *See* PIT recovery
- problems
 - areas to check 178

- profile
 - authenticating 69
 - creating from CLI 69
 - creating from GUI 72
 - defined 5
 - deleting from CLI 71
 - deleting from GUI 78
 - displaying backup list from GUI 78
 - explained 17
 - information needed to create 67
 - modifying from CLI 70
 - modifyng from GUI 77
 - properties dialog 77
- profile create operation
 - CLI command 154
 - CLI example 158
 - command parameters 154
- profile delete operation
 - CLI command 159
 - command parameters 159
 - example 160
- profile list operation
 - CLI command 161
 - CLI example 161
 - command parameters 161
- profile show operation
 - CLI command 162
 - CLI example 162
 - command parameters 162
- profile update
 - command parameters 164
- profile update operation
 - CLI command 164
 - CLI example 168
- properties dialog
 - profile 77
- protocol
 - license required 37
- protocols
 - accepted 11

Q

- qtrees
 - ASM issue with raw LUNs 185

- quiet command option
 - using 130

R

- RAC database
 - clone options 125
 - cloning 115, 125
 - setting up clone 126
 - specify one node 18
 - using with SnapManager 9
- Real Application Clusters *See* RAC
- recover
 - defined 5
 - option
 - general steps 91
- recover option
 - auto 5
 - command parameters 169
 - example 172
 - point-in-time 5
 - recovering data 90
 - used with restore 90
 - using with GUI 92, 94
 - using with restore command 92
- recoverable events 33
- Recovery Management *See* RMAN
- recovery operations
 - types 21
- redolog
 - clone map file key 118
- remove operation
 - CLI command 147
 - CLI example 149
 - command parameters 147
 - executing from CLI 105
 - executing from GUI
 - GUI
 - executing remove operation 105
 - general steps 104
- Reports pane
 - GUI 65
- repositories
 - multiple 19

- repository
 - authenticating 63
 - backing up 19
 - defined 5
 - loading the schema 49
- requirements
 - FlexClone 37
 - JRE 37
 - NetApp Host Agent 37
 - SnapDrive for UNIX 37
- respository
 - explained 16
- restore operation
 - CLI command 169
 - CLI example 172
 - command parameters 169
 - excuting from CLI 92
 - excuting from GUI 92
 - general steps 91
 - options 90
 - types 21
- retaining backups
 - examples 30
- RMAN
 - problem seeing backups 185
 - using with SnapManager 9
- rotating
 - log files 112
- rpm file
 - Linux installation 42

S

- shell
 - using pfile to enter commands 128
- SID
 - duplicate fails 183
- SIDs
 - used with clones 115
- sliding window 30
- SnapDrive for UNIX
 - required 11, 37
- SnapManager
 - additional documentation 13
 - backing up data 82

- capabilities 7
- checking the version 111
- compatibility matrix 10
- downloading software 41
- encrypted files 9
- executing 10
- getting 40
- how it works 9
- installation steps on Linux 42
- installing GUI 52
- licenses 37
- limitations 12
- loading the repository 49
- log files 112
- operational states 32
- prerequisites 10
- prerequisites 37
- running from GUI 64
- starting from CLI 62
- starting from GUI 63
- terms defined 4
- uninstalling from Linux host 44
- uninstalling from Solaris host 47, 48
- uninstalling the GUI 52
- upgrading versions 53
- working with RAC 18
- SnapManager Compatibility Matrix 13
- SnapManager Release Notes 13
- SnapRestore
 - required 37
- snapshot
 - defined 6
- software
 - downloading 41
 - uncompressing on Solaris 45
- Solaris host
 - .pkg file 45
 - example of uninstalling SnapManager 48
 - uncompressing SnapDrive for UNIX software 45
 - uninstalling SnapManager 47
- SQLPlus
 - confirming it runs 79
- storage manager
 - using SnapManager 34

System Configuration Guide 14

system setup

- verifying from CLI 79

- verifying from GUI 79

T

- tablespace 7

Task pane

- GUI 66

terms

- auto recovery 5

- backup 4

- GUID 4

- offline 5

- online 5

- PIT recovery 5

- profile 5

- recover 5

- repository 5

- snapshot 6

troubleshooting

- ASM backup fails 185

- backup fails without message 186

- backup hangs 186

- clone mount point issue 183

- clone needs unique SID 183

- controlfile recovery slow 185

- file system 186

- NID-00100 error 183

- problem seeing RMAN backups 185

- questions to ask 178

- unable to display GUI backups 182

- using the dump command 179

- verify operation fails 182

U

udump

- clone map file key 117

uninstalling

- SnapManager from Linux host 44

- SnapManager from Solaris host 47

- SnapManager GUI 52

unmount operation

- CLI command 150

- CLI example 151

- command parameters 150

- executing from CLI 108

- executing from GUI 109

- general steps 107

- unrecoverable events 33

unsuccessful backup

- deleting 30

upgrading

- SnapManager 53

V

verbose command option

- using 130

verify operation

- CLI command 173

- CLI example 174

- command parameters 173

- excuting from CLI 88

- excuting from GUI 89

- fails 182

- steps 88

version

- checking SnapManager version 111

version command

- CLI command 175

- CLI example 175

volume manager

- accepted 11

W

- warning messages 130

Windows system

- installing GUI 52

- uninstalling SnapManager GUI 52