SnapManager® 2.1 for Microsoft® SQL Server™
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 215-01752_A0
August 2006

Copyright and trademark information

Copyright information

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

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

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

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

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

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

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

Trademark information

NetApp, the Network Appliance logo, the bolt design, NetApp–the Network Appliance Company, DataFabric, Data ONTAP, 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 xi

Chapter 1 **Understanding SnapManager** 1

Overview of SnapManager for Microsoft SQL Server 2

Terms and technologies. 9

How SnapManager for Microsoft SQL Server works 13

Chapter 2 **Preparing to Install or Upgrade SnapManager** 19

What to read 20

Verifying Windows host system requirements 22

Verifying storage system requirements. 31

Preparing a Windows host system for SnapManager installation 33

Chapter 3 **Installing or Upgrading SnapManager** 35

Installing SnapManager on a stand-alone Windows host system 36

Installing SnapManager into an MSCS environment. 47

Upgrading to SnapManager 2.1 51

Uninstalling SnapManager 58

Reinstalling SnapManager 65

Chapter 4 **Starting SnapManager for the First Time** 67

Chapter 5 **Understanding Volume Mount Point** 73

Understanding NTFS volume mount point 74

Understanding SnapManager support for volume mount points 76

Backup and recovery using volume mount point. 82

Chapter 6 **Preparing to Migrate SQL Server Databases** 85
SQL Server configuration rules with SnapManager 86
SQL Server configurations supported with SnapManager 89
Assessing volume size 96

Chapter 7 **Using the SnapManager Configuration Wizard** 101
How databases are stored on storage system volumes 102
Understanding the Configuration Wizard 103
Migrating SQL Server databases to LUNs 107
Moving multiple SnapInfo directories to a single SnapInfo directory 123
Migrating SQL Server databases back to local disks. 126

Chapter 8 **Understanding SnapManager Backup Sets** 129
How SnapManager Backup works 130
Types of backup operations performed using SnapManager. 132
How SnapManager checks database integrity in backup sets 135
How SnapManager backup data is organized 139
Ways to manage the number of backup sets kept online 144
When to run a SnapManager backup 147

Chapter 9 **Backing Up Databases Using SnapManager** 149
How SnapManager backup functions are accessed. 150
Creating a full database backup using SnapManager. 152
Creating a transaction log backup using SnapManager 172
What to do if a SnapManager backup operation fails 184
Performing database verification using SnapManager 187
Using backup management groups in backup and verification. 198
Explicitly deleting backup Snapshot copies using SnapManager 202

Chapter 10 **Restoring Databases Using SnapManager** 213
SQL Server 2000 and 2005 recovery models. 214
Understanding SnapManager Restore 216
Types of SnapManager restore operations 222
Choosing the type of restore operation to perform 224
Performing a restore operation 225
Managing databases mounted at or restored to alternate locations. 242
Deleting restored Snapshot copies 247

Chapter 11 **Managing SnapManager Operational Reports** 249
Understanding the SnapManager Reports tab 250
Viewing a SnapManager report. 251
Printing a SnapManager report 252
Deleting SnapManager reports 253

Chapter 12 **Replicating Backups to Mirrored Volumes** 255
Understanding SnapManager backups with SnapMirror updates 256
Minimizing your exposure to loss of data 259
Scheduling SnapManager backups with SnapMirror replication. 263

Chapter 13 **Archiving SnapManager Backups** 267
Understanding SnapManager backup set archival 268
Choosing the best way to archive. 270
Archiving SnapManager backups using NDMP or dump 271
Archiving SnapManager backups using a Windows backup utility 274
Automatically archiving SnapManager backups using SnapVault. 278

Chapter 14 **Performing Disaster Recovery with SnapManager** 291
Guides you should read and have at hand 292
Preparing for disaster recovery 293

Backing up your SQL Server environment 295
Replicating your SQL Server environment 296
Restoring your SQL Server environment 298
Recovering SQL Server databases using SnapMirror 301
Recovering SQL Server databases using archives 309
Restoring a storage system volume using backups archived with SnapVault .314
Recovering a failed SQL Server computer 321
Recovering both a failed storage system and a failed SQL Server computer .324
Restoring databases from other SQL Server backups 326
Restoring system databases from SnapManager backup sets 337

Appendix A **Tools for Managing Backup and Verification** 339
Using the Run Command After Operation feature 340
Scheduling a backup job or a database verification job 349

Appendix B **SnapManager Command-Line Reference** 353
Guidelines for using the command-line utility 354
SmSqlBi syntax 356

Appendix C **Configuring SnapManager Application Settings** 363
Overview of SnapManager application settings 364
Connecting to an SQL Server instance 366
Changing the SQL Server user account 370
Database integrity verification options 372
SnapManager backup options 378
SnapManager restore options 382
Run Command After Operation default settings 385
Managing fractional space reservation 387
Enabling or disabling database migration back to local disks 398
SnapManager report directory options 399

Event notification options 401

Appendix D **Configuration Wizard Screens and Dialog Boxes** 407

Using the Database Verification Server and Account Authentication screen .408

Using the Where to Move Databases screen 410

Using the Number of Databases that Can Be Backed Up Concurrently screen 417

Using the Modify SQL Server Maximum Worker Threads Parameter screen .419

Using the Specify a Single SnapInfo Directory screen 423

Using the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen. 425

Using the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen 428

Using the Where to Put the SnapInfo Directory for Each Database screen . .431

Using the Database Migration Options screen 434

Appendix E **Configuring Post-Restore Database Recovery** 435

Understanding post-restore database recovery states 436

Specifying the post-restore state of a single database 437

Specifying the post-restore state of multiple databases 441

Index 447

Preface

About this guide

This guide describes how to install SnapManager® 2.1 for Microsoft® SQL Server™ software and how to migrate, back up, and restore SQL Server databases using SnapManager.

This guide does not explain how to install, configure, and use the SnapDrive® features that enable you to create and connect to LUNs, which you must do before you use SnapManager. For this information and details about how to configure storage system volumes and enable storage system access, see the *SnapDrive Installation and Administration Guide* for the version of SnapDrive that you are running.

This guide does not cover basic system or network administration topics, such as IP addressing, routing, and other network topology.

Audience

This guide is for system and database administrators who possess a working knowledge of SQL Server 2000 or 2005 and Windows® 2000 Server or Windows Server 2003. The discussion assumes familiarity with the following topics:

- SQL Server 2000 or 2005 administration
- Network functions and operations
- Basic concepts for moving data over a network
- Your operating system, network, and storage system
- FCP and iSCSI LUN access protocols

Is this the right guide for you?

Use the following table to determine which documents to read.

If you are...	Then read...
Installing storage systems in an Ethernet environment	The Data ONTAP® product documentation, which is delivered with the storage system.
Installing storage systems in a Fibre Channel environment	The document *Block Access Management Guide* for your version of Data ONTAP.

If you are...	Then read...
Installing SnapManager on a Windows host that is connected to a storage system and you plan to use iSCSI as the protocol for your LUN	◆ This document. ◆ If SnapDrive is not already installed, first read the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
Using the SnapMirror® feature in conjunction with SnapManager	◆ This document. ◆ The *Data ONTAP Data Protection Guide* for your version of Data ONTAP.

Typographic conventions

The following table describes the typographic conventions used in this guide.

Convention	Type of information
Italic type	Words or characters that require special attention. Placeholders for information you must supply. For example, if the guide says to enter the `arp -d` *hostname* command, you enter the characters "arp -d" followed by the actual name of the host. Book titles in cross-references.
`Monospaced` font	Command and daemon names. Information displayed on the system console or other computer monitors. The contents of files.
`Bold monospaced` font	Words or characters you type. What you type is always shown in lowercase letters, unless you must type it in uppercase letters.

Special messages

This guide contains special messages that appear in the following circumstances:

Note

A note provides important information to help you install or operate your system efficiently.

Attention

An attention contains instructions that keep you from crashing your system or losing data.

Understanding SnapManager 1

This section provides an overview of how SnapManager 2.1 for Microsoft SQL Server works. The following topics are covered:

- "Overview of SnapManager for Microsoft SQL Server" on page 2
- "Terms and technologies" on page 9
- "How SnapManager for Microsoft SQL Server works" on page 13

Overview of SnapManager for Microsoft SQL Server

SnapManager provides an integrated data management solution for SQL Server 2000 or 2005 that dramatically boosts the availability and reliability of SQL Server databases. This section explains briefly what SnapManager does and does not do and describes its components. See the following topics for more information:

- "New functionality with SnapManager 2.1" on page 2
- "What SnapManager does" on page 3
- "What SnapManager does not do" on page 5
- "Where you install and run SnapManager" on page 6
- "How you use SnapManager" on page 6
- "About the SnapManager graphical user interface" on page 7
- "About the SnapManager command-line interface" on page 8

New functionality with SnapManager 2.1

SnapManager 2.1 for Microsoft SQL Server supports the following new features:

- Fractional space reservation policy settings
- NTFS volume mount point support
- Available both in x86 and x64 builds

Fractional space reservation policies: If a storage system volume is configured with a fractional space reservation of less than 100 percent, write operations to LUNs on that volume are not guaranteed unless the reserve level is monitored, and preventive action is taken when overwrite reserve utilization is too high. SnapManager 2.1 for SQL provides an SQL-aware space management tool, designed specifically for LUNs that reside on volumes with fractional space reservation set to less than 100 percent for volumes that store SQL databases or SnapManager SnapInfo directories.

NTFS volume mount point support: NTFS volume mount point support enables SnapManager 2.1 for SQL to manage SQL databases stored on mounted NTFS volumes (on dedicated disks or shared cluster resource disks) or standard Windows volumes. Volume mount point enables you to surpass the 26 drive letter limitation.

What SnapManager does

SnapManager provides rapid online backup and near instantaneous restoration of SQL Server 2000 and 2005 databases by using the online Snapshot™ technology that is part of Data ONTAP®. SnapManager is not a VSS requestor. It uses the Virtual Device Interface (VDI) API to manage SQL Server backup and restore operations.

Data management: SnapManager supports the following data management capabilities:

- Migrating SQL Server databases to LUNs on storage systems
- Backing up SQL Server databases from LUNs on storage systems
- Verifying the backed-up SQL Server data files
- Verifying the backed-up SQL Server transaction log files during backup
- Managing the SnapManager backups
- Restoring the SQL Server databases from previously created SnapManager backups

Data archival for long-term or remote storage of backups: You can use SnapManager to create offline archives of Snapshot copies containing SQL Server backup sets. Three different archive methods are supported:

- Manually initiated archival using NDMP (Network Data Management Protocol) or the storage system's `dump` command
- Manually initiated archival using a Windows backup utility
- Automatic archival using the Run Command After Operation feature with your backup operation

  The SnapVault software can be used to archive backup sets.

The archival of SQL Server database backup sets using SnapManager is described in Chapter 13, "Archiving SnapManager Backups," on page 267 and in Chapter 14, "Performing Disaster Recovery with SnapManager," on page 291.

Data replication for disaster recovery: When used with SnapMirror, SnapManager provides the ability to automatically replicate databases stored on the source volume to its mirrored target volume situated locally or remotely. Data replication using SnapManager with SnapMirror is described in Chapter 12, "Replicating Backups to Mirrored Volumes," on page 255 and in Chapter 14, "Performing Disaster Recovery with SnapManager," on page 291.

What SnapManager does not do

SnapManager 2.1 for Microsoft SQL Server does not support the following uses:

- SnapManager does not support Microsoft SQL Server 6.5 or 7.0.
- SnapManager does not support SnapDrive versions prior to 4.1.
- SnapManager does not create or restore backups of Microsoft SQL Server databases that are stored on storage devices that are provided by companies other than NetApp.
- SnapManager does not support filegroup backups or filegroup restores of Microsoft SQL Server databases.
- SnapManager is not capable of backing up or restoring SQL Server databases that are accessed through CIFS.

Where you install and run SnapManager

You can install and run SnapManager on your SQL Server computer or on a different Windows host system.

Local administration: When you run SnapManager on your SQL Server computer, it is called *SnapManager local administration*. Typically, you install SnapManager on the same Windows host system as your SQL Server computer. System requirements for a SnapManager local administration are described in Chapter 2, "Preparing to Install or Upgrade SnapManager," on page 19.

Remote administration: When you run SnapManager on a different Windows network host, it is called *SnapManager remote administration*. If you install SnapManager on a computer remote from the SQL Server computer, you can run SnapManager remotely to perform any task that you can on a locally installed SnapManager system.

System requirements for a SnapManager remote administration system are described in Chapter 2, "Preparing to Install or Upgrade SnapManager," on page 19.

Remote verification: From a remote administration server that is configured with SnapDrive and SQL Server, you can also perform remote database verification. Remote verification offloads the CPU-intensive database verification operations that can affect the performance of your production SQL Server computer.

System requirements for a SnapManager remote administration system used for remote verification are described in Chapter 2, "Preparing to Install or Upgrade SnapManager," on page 19.

The setup and use of a remote verification server is described in Chapter 9, "Backing Up Databases Using SnapManager," on page 149.

How you use SnapManager

The following steps describe a typical way to use SnapManager:

1. After installing SnapManager, use the SnapManager Configuration Wizard to dismount your SQL Server databases and move them to LUNs on a storage system. The Configuration Wizard ensures that your SQL Server data is placed correctly.
2. Use SnapManager Backup to create backups of the data.
3. If necessary, you can use SnapManager Restore to restore your SQL Server databases from one of the backups.

4. Using SnapManager's backup facility to initiate SnapMirror through SnapDrive, you can create mirror replications of these databases to be used for various purposes, such as disaster recovery.

About the SnapManager graphical user interface

The SnapManager graphical user interface enables you to manage backup and restore operations of SQL Server database files and transaction log files on LUNs. The user interface also enables you to schedule backups and verify the integrity of databases in SnapManager backup sets. You can also use the SnapManager user interface to administer SnapManager for SQL Server on another SQL Server computer on the network. SnapManager contains the following components:

SnapManager Configuration Wizard: Use the Configuration Wizard to detach your SQL Server data and log files and move them from the local current disk to the selected LUNs. It also reattaches the databases and creates SnapInfo directories for storing information about backed-up data sets and logs. The Configuration Wizard also ensures a correct configuration for performing successful backup and restores. More detailed information is provided in Chapter 6, "Preparing to Migrate SQL Server Databases," on page 85 and in Chapter 7, "Using the SnapManager Configuration Wizard," on page 101.

SnapManager Backup Wizard: Use either the Backup Wizard or the Backup and Verification tab to perform and schedule online full data and transaction log backups. SnapManager Backup also allows the database verification option to be run as part of the backup operation. More detailed information is provided in Chapter 8, "Understanding SnapManager Backup Sets," on page 129 and in Chapter 9, "Backing Up Databases Using SnapManager," on page 149.

SnapManager Restore Wizard: Use either the Restore Wizard or the SnapManager Restore tab for rapid recovery of an SQL Server database. The SnapManager Restore Wizard guides your selection of the particular backup to restore. SnapManager also enables you to replay transaction log backups, which restore your SQL Server database state. More detailed information is provided in Chapter 10, "Restoring Databases Using SnapManager," on page 213.

SnapManager Reports tab: Use the SnapManager Reports tab to access the operational reports that SnapManager automatically creates for all log files, database configurations and migrations, deleted Snapshot copies, backups, and restores. Each report is essentially a history file that includes step-by-step details of the operation, the final status of the operation, and error messages for any errors encountered during the operation. More detailed information is provided in Chapter 11, "Managing SnapManager Operational Reports," on page 249.

About the SnapManager command-line interface

The SnapManager command-line interface enables you to initiate a backup or verification job. The command-line interface supports all of the options available through the Backup and Verification tab or the Backup Wizard in the SnapManager graphical user interface.

Terms and technologies

This section defines the terms and technologies referenced in this guide. Each term or technology is described within a SnapManager-specific context. See the following topics for more information:

- "backup set" on page 9
- "cluster group" on page 9
- "database consistency checker (DBCC)" on page 9
- "filegroups" on page 10
- "host system" on page 10
- "log shipping" on page 10
- "MSCS" on page 10
- "multiple-instance cluster" on page 10
- "quorum disk" on page 10
- "recovery model" on page 10
- "single-instance cluster" on page 11
- "storage system" on page 11
- "SQL Server 2000 or 2005" on page 11
- "SQL Server computer" on page 11
- "system database" on page 11
- "transaction log" on page 12
- "user database" on page 12

backup set

All the data that SnapManager for Microsoft SQL Server requires to perform an individual restore operation.

cluster group

A logical group of cluster resources that can be moved from one node to the other while the nodes remain operational. The cluster group can be moved by the administrator, or it can be moved as a result of a cluster resource failure.

database consistency checker (DBCC)

The Microsoft SQL Server utility for finding and correcting problems in the consistency of the database.

filegroups

A group of files in a user database that have a common logical identity.

host system

A computer that accesses storage on a storage system.

log shipping

A process that takes transaction logs from a primary SQL Server and applies them sequentially on a scheduled basis to another SQL Server. In the event of a failure, an application could be redirected to the other server, which would be only slightly unsynchronized with the primary server or potentially up to date with it, depending on when the last transactions were applied. Log shipping is a means of protecting organizations in the event of a logical or physical system failure.

MSCS

Microsoft Cluster Services. Software that can be added to an SQL Server computer to enable an SQL Server system to operate in standby mode for the primary SQL Server system. SnapManager 2.1 for Microsoft SQL Server supports two-node clusters in a Windows 2000 Advanced Server environment and up to four-node clusters in a Windows Server 2003 Enterprise Edition environment. See also “cluster group” on page 9.

multiple-instance cluster

(Active/Active mode) refers to an MSCS cluster in which two or more SQL Server instances are running on different nodes in the same cluster. Each instance has its own disk resources that cannot be used by other instances. If one system fails, the other system takes over its application services. See also “single-instance cluster” on page 11.

quorum disk

A shared disk resource used by MSCS to keep track of cluster management information, such as cluster resources and state. The quorum disk should not be used for SQL Server files. When SQL Server databases are migrated to LUNs for use with SnapManager, a separate LUN is required for the quorum disk.

recovery model

There are three distinct ways that you can recover your SQL Server environment in the event of a failure. Each model addresses a different need for performance, disk and tape space, and protection against data loss. The three models are summarized as follows:

SIMPLE: You recover the most recent full backup and any differential backups that might exist. Data can be recovered only to the last backup.

FULL: All operations are logged, including bulk operations and bulk loading of data.

BULK_LOGGED: Certain database operations (including SELECT INTO, BULK COPY/BCP, CREATE INDEX, WRITETEXT, and UPDATETEXT) are logged minimally or not at all. Because the BULK_LOGGED recovery model does not provide point-in-time recovery, the risk of data loss is higher than with the FULL recovery model.

For more information, see your Microsoft SQL Server documentation.

single-instance cluster

(Active/Passive mode) refers to an MSCS cluster with SQL Server installed, where only one active instance of SQL Server is owned by a node and all other nodes of the cluster are in a standby state. See also Appendix , "multiple-instance cluster," on page 10.

storage system

Any NetApp storage product that supports FCP (Fibre Channel Protocol) and iSCSI (SCSI over IP).

This guide refers to all NetApp storage products (filers, FAS appliances, and NearStore systems) as *storage systems*. In other documentation, the terms *filer* and *appliance* have also been used to refer to storage systems.

SQL Server 2000 or 2005

A Microsoft relational database system based on the client/server database model.

SQL Server computer

The hardware on which a Microsoft SQL Server 2000 or 2005 database system is running.

system database

A set of five databases created when you install Microsoft SQL Server 2000 or 2005. See also "user database" on page 12. The five system databases are described as follows:

distribution database: A database on the distributor that stores data for replication, including transactions, Snapshot jobs, synchronization status, and replication history information. This is visible, only visible if replication is turned on.

master database: Records the system-level information, SQL Server initialization information, and configuration settings for SQL Server. This database also records all login accounts, the existence of all other databases, and the location of the primary file for all user databases.

tempdb database: Holds temporary tables and temporary stored procedures. This database is also used for other temporary storage needs of SQL Server, such as for sorting data. A clean copy of the tempdb database is re-created at its default size every time SQL Server is started.

model database: Serves as a template for all other databases on the system, including the tempdb database. When a database is created, the first part of it is created as a copy of the contents of the model database. The rest of the database is filled with empty pages. The model database must exist on the system because it is used to re-create tempdb every time SQL Server is started. You can alter the `model` database to include user-defined data types, tables, and so on. If you alter the `model` database, every database you create will have the modified attributes.

msdb database: Holds tables that SQL Server Agent uses for scheduling jobs and alerts and for recording operators (those assigned responsibility for jobs and alerts). This database also holds tables used for log shipping and for backup and recovery.

transaction log

A file in the database that is used to record all modifications to the database. Information about the transactions that perform modifications is also kept in the transaction log. The transaction log stores enough information to undo a transaction (in case a rollback is necessary) or to redo a transaction (in case a transaction needs to be rolled forward).

user database

A database created by an SQL Server user and used to store application data. Most users connecting to instances of SQL Server reference user databases only, not system databases. See also “system database” on page 11.

How SnapManager for Microsoft SQL Server works

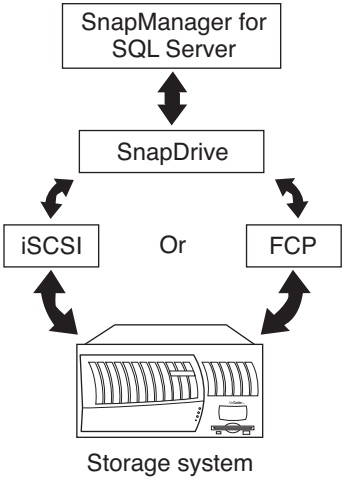
This section describes how SnapManager works with the other system components. See the following topics for more information:

- "System overview" on page 13
- "How SnapManager and SnapDrive work together" on page 15
- "How SnapManager works with other backup methods" on page 16
- "How SnapManager works with Snapshot copies used as backups" on page 17
- "How the storage system safeguards data" on page 18

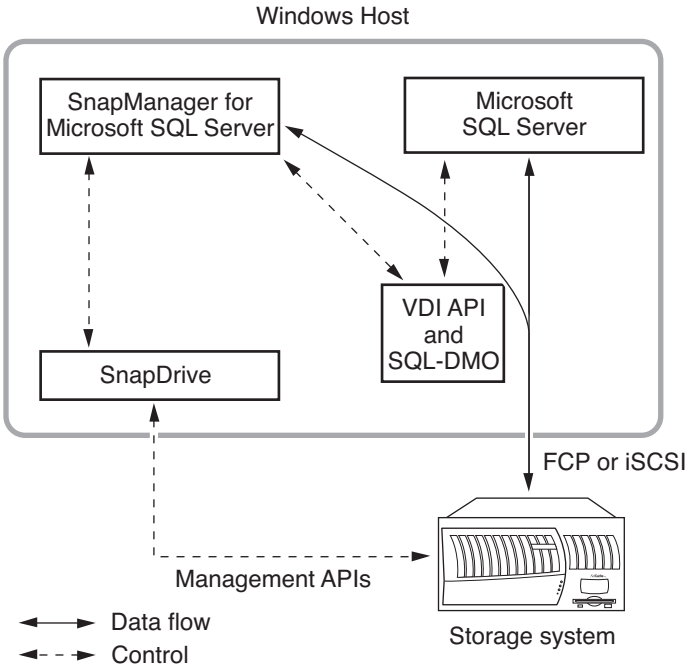
System overview

SnapManager is an SQL Server 2000- and 2005-aware application that provides backup and restore functionality in an SQL Server environment.

Relationship with other components of an SQL Server installation backed by a storage system: The following illustration shows the relationship between storage systems, SnapDrive, and SnapManager 2.1 for Microsoft SQL Server.



Relationship with SQL Server components: SnapManager 2.1 for SQL Server is installed on an SQL Server computer running Windows 2000 Server or Windows Server 2003. SnapManager is an SQL Server 2000- and SQL Server 2005-aware application that uses the SQL Server *VDI API* and the SQL Distributed Management Objects *(SQL-DMO)* to provide backup and restore functionality in an SQL Server environment, as shown in the following illustration.



SnapManager on Windows Server

How SnapManager and SnapDrive work together

SnapDrive integrates with the NTFS Windows file system and the Windows Volume Manager to enable the management of the LUNs on a storage system in a Windows environment. SnapDrive provides the underlying layer of support for SnapManager by making these LUNs available as local disks on the Windows host system.

When to use SnapManager: Having SnapDrive installed on the SQL Server is a requirement for using SnapManager. After SnapDrive is installed and your LUNs are created, you can use SnapManager to migrate SQL Server databases from local disk to LUN and perform backups and restores using Snapshot technology.

When to use SnapDrive: SnapDrive manages LUNs on a storage system, making these disks appear available and as ordinary local disks on the Windows host server. This enables the server to interact with the LUNs as if they were directly attached, physical disks. You can use SnapDrive to create, connect, expand, and manage LUNs. For information about how to perform these tasks using SnapDrive, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

How SnapManager works with other backup methods

NetApp recommends that you employ SnapManager technology as a complement to conventional backup processes.

Supplementary backup archive: SnapManager backups are not intended to replace data archiving schemes in place for long-term or permanent data retention. Because SnapManager backups reside on primary disk, you should move your data to alternative media locations, such as a NearStore system or some other type of secondary storage media. NDMP or the storage system `dump` command are the most efficient methods for creating archives of the LUN drive files.

Enterprise Manager or Management Studio backup utility: Because some types of third-party backup applications truncate transaction logs and interfere with the SnapManager for SQL Server recovery process, you should not perform transaction log backups with any application other than SnapManager for SQL Server.

What SnapManager does not back up: SnapManager does not back up all the files commonly used by an SQL Server computer. You can use NTBackup (a native backup utility that ships with Windows) to back up the system state and the file systems on hard disks connected to the SQL Server.

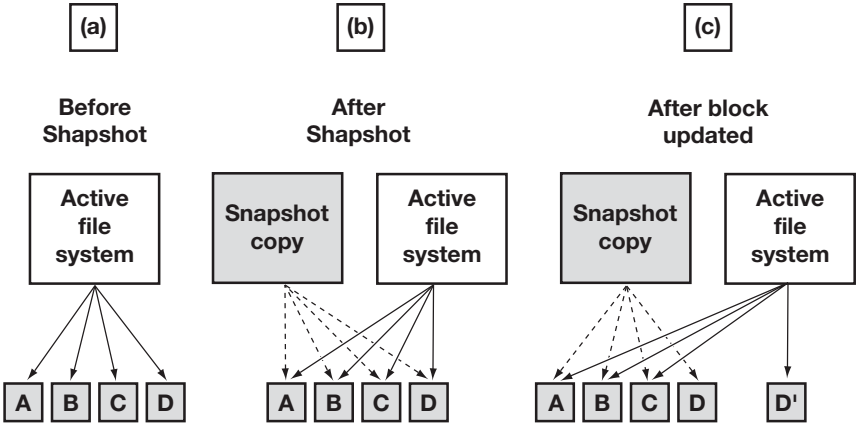
Note

You can use NTBackup to archive SnapManager backup sets to a file, instead of using tape, and store that file on a storage system.

How SnapManager works with Snapshot copies used as backups

A Snapshot copy is a point-in-time, read-only copy of an SQL database stored on a storage system volume. SnapManager Backup uses Snapshot functionality to create real-time, online, read-only copies of the SQL database. Snapshot copies enable you to restore your SQL database rapidly if you encounter data corruption or other problems.

The following illustration provides an example of how Snapshot copies work.



Example: Suppose you make a Snapshot copy of a file named file.txt that spans four disk blocks in the active file system. Initially, the Snapshot version of file.txt and the version in the active file system are identical: the same blocks on disk store both versions, so the Snapshot copy version of file.txt consumes no additional disk space.

Now suppose that you make a modification to file.txt that affects only one of the four disk blocks. The new data cannot overwrite the original block because that block is needed as part of the Snapshot copy. As a result, the new data is written to a new disk block and the file's inodes are updated accordingly. The active file system inodes now reference the three original disk blocks that have not been modified since the Snapshot copy, plus the one new block. The Snapshot copy inodes still reference the original four blocks.

If you delete file.txt, the blocks holding its data are no longer part of the active file system, but they are still part of the Snapshot copy; deleting file.txt from the active file system does not free any disk space until the Snapshot copy is deleted.

How the storage system safeguards data

Under SnapManager, the SQL Server data and log files reside on a LUN created on the storage system and formatted with the New Technology File System (NTFS).

The storage system is a volume manager that stores LUNs. The storage systems use battery-protected nonvolatile RAM (NVRAM) to protect incoming file system I/O operations. The contents of NVRAM are flushed to disk at regular intervals—more frequently if the NVRAM fills up, even during periods of inactivity. This ensures that the file system is always in a consistent state. The storage system guarantees that the contents of NVRAM are always written to disk, even during a power failure.

Preparing to Install or Upgrade SnapManager 2

This section describes the tasks that you must complete prior to installing or upgrading SnapManager 2.1 for SQL Server.

Before you begin installing or upgrading to SnapManager 2.1 for SQL Server, ensure that you complete the prerequisite tasks described in this section:

- "What to read" on page 20
- "Verifying Windows host system requirements" on page 22
- "Verifying storage system requirements" on page 31
- "Preparing a Windows host system for SnapManager installation" on page 33

What to read

In addition to this guide, you should be familiar with the following guides and have them available for reference during installation, upgrade, or administration. All of the NetApp product documents can be found on the NetApp on the Web (NOW) site, which is located at http://now.netapp.com/NOW/knowledge/docs.

Reference documents

You should have the following documents readily available for reference.

For detailed information about...	Read...
Configuring your storage system	*SnapDrive Installation and Administration Guide* for your version of SnapDrive
Configuring your Windows host system	Microsoft Windows documentation
Configuring SQL Server 2000 or 2005 on your Windows host system	Microsoft SQL Server 2000 or 2005 documentation
Installing or upgrading SnapDrive	*SnapDrive Installation and Administration Guide* for your version of SnapDrive

Current information

The following documents contain the most current information about NetApp products.

For current information about...	Read...
Compatible versions of SnapDrive, SnapManager, and Data ONTAP	*SnapManager and SnapDrive Compatibility Matrix*
SnapDrive system requirements: ◆ Operating system ◆ Hardware platform ◆ Network	*SnapDrive NOW Description page* for your version of SnapDrive
SnapManager for SQL Server system requirements	*SnapManager 2.1 for Microsoft SQL Server NOW Description page*

Verifying Windows host system requirements

In the most basic configuration, SnapManager is installed on the same Windows host system as Microsoft SQL Server 2000 or 2005. In addition to this, you can install SnapManager on one or more remote Windows hosts for remote administration of the SQL Server computer or for remote verification of the databases contained in SnapManager backup sets.

The following topics describe the system requirements for all of the previously listed configurations of SnapManager:

- "Windows host system requirements" on page 23
- "SnapManager license requirement" on page 26
- "SnapManager service account requirements" on page 26
- "SnapManager service account requirements in workgroup mode" on page 27
- "SnapDrive requirement" on page 27
- "Requirements for a remote administration server" on page 28
- "Requirements for a remote verification server" on page 29

Windows host system requirements

To run SnapManager, you must meet the following requirements.

Windows host component	Requirement if the host system is running Windows 2000 Server	Requirement if the host system is running Windows Server 2003
Operating system	One of the following: ◆ Windows 2000 Server with SP4 ◆ Windows 2000 Advanced Server with SP4 If the host system is running SQL Server 2005, the following version of Microsoft Data Access Components (MDAC) must be installed: ◆ MDAC 2.8 SP1	One of the following: ◆ Windows Server 2003 Standard Edition with SP1, 32-bit edition ◆ Windows Server 2003 Enterprise Edition with SP1, 32-bit edition **Note** Windows Server 2003 SP1 includes MDAC 2.8 SP2, which is required for SQL Server 2005 on Windows Server 2003. ◆ Microsoft Windows Server 2003 R2 Enterprise x64 Edition ◆ Microsoft Windows Server 2003 R2 Enterprise x86 Edition
Microsoft Windows hotfixes	See the SnapDrive software system requirements.	See the SnapDrive software system requirements.

Windows host component	Requirement if the host system is running Windows 2000 Server	Requirement if the host system is running Windows Server 2003
Microsoft SQL Server	One of the following editions of Microsoft SQL Server: ◆ Microsoft SQL Server 2000 ❖ Standard Edition with SP3a or SP4 ❖ Enterprise Edition with SP3a or SP4 ◆ Microsoft SQL Server 2005 ❖ Standard Edition ❖ Standard Edition with SP1, x86 or x64 versions ❖ Enterprise Edition ❖ Enterprise Edition with SP1, x86 or x64 versions **Note** The preceding requirement does not apply to a remote administration server.	
SQL Browser service	If the host system is running SQL Server 2005, configure the SQL Browser service to start automatically.	
SnapDrive	SnapDrive 4.1 **Note** For a remote *administration* server, SnapDrive is optional unless you intend to use the remote administration server to remotely administer SnapDrive. For a remote *verification* server, SnapDrive is required.	
Windows registry keys	When used with SnapDrive 4.1 in a clustered environment, SnapManager 2.1 is capable of restoring databases in the cluster without taking the SQL Server cluster group offline. To enable this feature, you must configure all of the nodes in the cluster by setting a registry variable. This configuration step is described in the procedure "Installing SnapManager and creating a new Windows cluster" on page 49.	

Windows host component	Requirement if the host system is running Windows 2000 Server	Requirement if the host system is running Windows Server 2003
Licenses enabled	Depends on the type of SnapManager licensing purchased: ◆ If SnapManager is licensed per SQL Server, a SnapManager license is required on the Windows host system. ◆ If SnapManager is licensed per storage system, a SnapManager license is required on the storage system.	
Other hardware and software	See your SnapDrive documentation for complete details about the following system requirements: ◆ Host hardware platform ◆ LUN access protocol (FCP or iSCSI) software **Note** The preceding requirements do not apply to a remote administration server.	

Note

For the most current information about the Windows host system requirements, see the *SnapManager 2.1 for Microsoft SQL Server Description* page at the NetApp on the Web (NOW) site (http://now.netapp.com/NOW/software).

SnapManager license requirement

SnapManager 2.1 for Microsoft SQL Server requires that a SnapManager license be enabled on either the SQL Server or on the storage system.

Per-SQL Server license: This license is for a specific SQL Server and provides capacity-based utilization for enterprise accounts. This type of license is also called a *per-server license or a server-side license*. With server-side licensing, no SnapManager license is required on the storage system.

Note

If you are using a per-server SnapManager license, you can enter the license key while you are installing the SnapManager software (in the Customer Information screen of the SnapManager software installation utility), or you can enter the license key later (from the About SnapManager dialog box), after SnapManager is installed.

Per-storage-system license: This license is for a specific storage system and enables you to use SnapManager on the storage system with any number of SQL Server instances and any database sizes. This type of license is also called a *per-filer license or a filer-side license*. If no server-side license is detected, SnapManager checks the storage system for a filer-side license when a SnapManager operation is initiated. If a filer-side license is not enabled, the SnapManager operation fails and an error is written to the Windows event log.

SnapManager service account requirements

In order to run the SnapManager Backup and Restore functions, the SnapManager service account must have *sysadmin fixed server role* privileges on the SQL Server. This is usually addressed by giving the Windows user account administrator rights on the SQL Server. You can meet this requirement by adding the Windows domain account used by SnapManager to the system administrator's server role on the SQL Server.

SQL Server authentication: When using SQL Server authentication, the SQL Server administrator must have *sysadmin server role privileges on the SQL Server instance*. SnapManager for SQL Server requires that the database administrator have the required privileges to mount and dismount databases, back up data and transaction log files, and restore those same files. The system administrator role fulfills all these permissions requirements.

Windows authentication: When using Windows authentication, the Windows account that you are logged onto the system with must have *system administrator privileges on the SQL Server.* This account is presumably the same account running SnapDrive. However, some organizations give different categories of administrators different responsibilities and therefore different levels of access. For example, one group of administrators might run

SnapManager to manage the SQL Server databases while another group of administrators might run SnapDrive to manage the LUNs. In this case, separate accounts would be used for SnapDrive and SnapManager. The SnapManager server would still run under the SnapManager user account.

Note

A storage system administrator must perform and verify the configuration.

SnapManager service account requirements in workgroup mode

To use SnapManager with Windows in *workgroup mode*, the SnapManager service account must be a *local user account* (not a domain account) that meets the requirements described in "SnapManager service account requirements" on page 26. For instructions on how to configure SnapDrive in workgroup mode, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

SnapDrive requirement

In a SnapManager configuration, SnapDrive is required on all Windows host systems except those that are used exclusively as remote administration servers.

On the Microsoft SQL Server computer: In order to use SnapManager to migrate, back up, and restore SQL Server data, you must have SnapDrive installed on the same machine and use it to create two or more dedicated LUNs to hold the SQL Server data and log files and the SnapManager SnapInfo directory (the SnapInfo directory and the transaction log files cannot be housed on the same LUN).

Note

NetApp recommends that you create a minimum of three LUNs so that the data files for system databases and user databases can be placed on separate LUNs.

For a remote administration server: The SnapManager installation package enables you to install SnapManager on a Windows host system without SnapDrive. This enables you to install SnapManager on a remote system that will be used to administer the Microsoft SQL Server computer that is running with SnapManager. As long as this remote system will not be used to perform database verification, SnapDrive is not required.

For a remote verification server: You can also install SnapManager on a remote Windows host system for the purpose of offloading the CPU-intensive database verification process. SnapDrive must be installed on this remote system, but you do not need to have LUNs on that system. For installation details, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

Implications for your storage system: The storage system used in your SnapManager environment must also be configured to meet the SnapDrive license requirements. For information about compatible versions of SnapManager, SnapDrive, and Data ONTAP, see the *SnapManager and SnapDrive Compatibility Matrix*. You can access this information from the NetApp on the Web (NOW) site, which is located at http://now.netapp.com/NOW/knowledge/docs. The system requirements for a storage system used in a SnapManager environment are described in "Verifying storage system requirements" on page 31.

Requirements for a remote administration server

An SQL Server that is running SnapManager can be remotely administered from another Windows system that is configured as a remote administration server:

- You do not need to install an iSCSI driver or an HBA driver on this system.
- SnapDrive does not need to be installed unless you want to use the remote administration server to remotely administer SnapDrive.
- SnapManager must be installed.

Note

Some limitations apply to using SQL Server authentication as the security authentication method to be used to establish the connection to a remote administration server. For more information, see "About SQL Server authentication" on page 366.

Requirements for a remote verification server

Remote verification is performed using the same mechanisms used for local verification, except that the verification is performed on a host that is different from the SQL Server that initiated the backup. This is the reason that you need SnapDrive and SnapManager installed on your remote verification server, as well as connectivity to the storage system (through FCP or iSCSI).

To run remote database consistency checks, your remote Windows system must meet the following requirements:

- The appropriate LUN driver (iSCSI or FCP) must be installed.
- The remote Windows system must have connectivity (iSCSI or FCP) to the storage system.
- SnapDrive must be installed.

Attention

Do not try to connect the SQL Server's LUNs to the remote SQL Server.

- SnapManager must be installed, but it does not need to be configured. You must specify the user account that you use for the production SQL Server.
- Microsoft SQL Server 2000 or 2005 must be installed.
- The SQL Server used for database verification cannot be a virtual SQL Server. However, a local SQL Server instance installed on one of the MSCS nodes can be used for database verification.

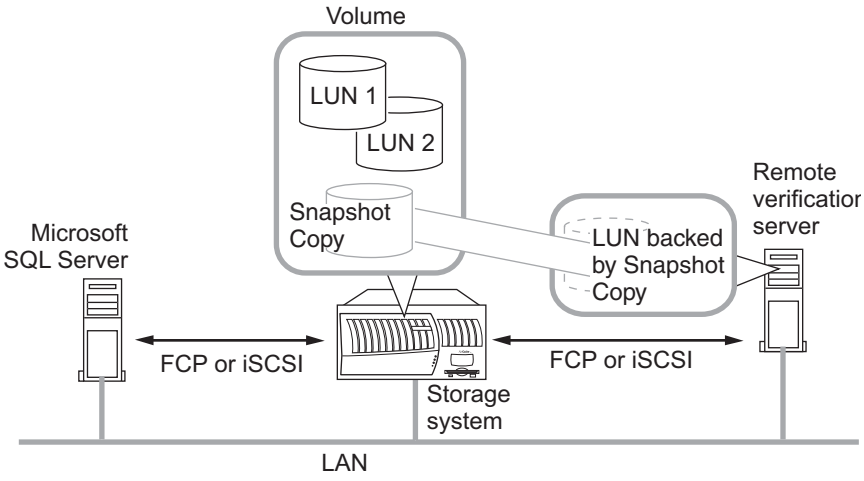
Note

If you cannot use a remote SQL Server instance or a local nonclustered SQL Server instance as the verification server, you can select not to perform database verification. Alternatively, you can select to verify only online databases before and after backup.

Attention

Although it is possible to restore from an unverified backup, NetApp strongly recommends that you restore only from verified backups.

The connectivity required to perform remote verification is shown in the following illustration.



Verifying storage system requirements

The following topics describe the system requirements for the storage systems used with SnapManager for Microsoft SQL Server:

- "Storage system requirements" on page 31
- "About the SnapManager license requirement" on page 31
- "About the additional requirements" on page 32

Storage system requirements

To be used with SnapManager, your storage system must meet the following requirements.

Storage system component	Requirements
Data ONTAP	See the SnapDrive software requirements described in the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
SnapManager license	A SnapManager license is required on the storage system only if you have licensed SnapManager on a per-storage-system basis.
Additional requirements	All of the SnapDrive requirements that apply: ◆ SnapRestore® feature ◆ iSCSI and/or FCP protocols ◆ SnapMirror license (if in use) ◆ Preferred IP address (if multiple IP addresses)

About the SnapManager license requirement

If a SnapManager for Microsoft SQL Server license is not enabled on the SQL Server computer, a SnapManager license must be enabled on the storage system.

After SnapManager has been installed, you can apply a server-side license key by clicking the About button in the task bar.

About the additional requirements

The storage system must also be configured to meet the SnapDrive requirements that apply.

SnapRestore: For SnapManager restore operations to work properly, the SnapRestore feature of SnapDrive must be licensed on the storage system that stores the SQL Server databases. For more information, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

iSCSI or FCP protocols: The appropriate LUN access protocol software must be installed on the storage system that stores the SQL Server databases. For more information, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

SnapMirror: If you plan to use the SnapMirror® software in conjunction with SnapManager, a SnapMirror license is required on both the source and target storage systems. For more information, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

SnapDrive preferred IP address: If your storage system has multiple IP addresses, configure the SnapDrive preferred IP address.

- For SnapDrive 3.1.1R1, see the NetApp Knowledge Base article KB4350 on the NOW site.
- For newer versions of SnapDrive, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

If you do not configure a SnapDrive preferred IP address for a storage system that has multiple IP addresses, SnapDrive times out when attempting to create Snapshot copies simultaneously for more than one virtual disk on a storage system.

Preparing a Windows host system for SnapManager installation

Before you install SnapManager on a supported Windows host system, complete the following steps.

Step	Action
1	Back up your system resources using your current backup tool or another industry standard backup utility. **a.** Back up the operating system installed on the SQL Server, including the system state, the boot and system drives, and the registry. **b.** Back up your SQL Server databases and transaction log files. **c.** Use the NTBackup utility that is part of the Windows operating system to create and maintain a current emergency repair disk (ERD).
2	Be sure you understand all the installation and configuration steps needed to make SnapManager work in your particular environment. ◆ If you plan to administer SnapManager for SQL server *locally* from this host, then proceed directly to Step 3. ◆ If you plan to administer SnapManager for SQL Server *remotely* from this host, review "Requirements for a remote administration server" on page 28 to determine your installation requirements. SnapDrive will not need to be installed unless you want to use the remote administration server to remotely administer SnapDrive. ◆ If you plan to use SnapManager for SQL Server on this host only to perform *remote verification*, then review "Requirements for a remote verification server" on page 29 to determine your installation requirements.
3	If you need to install SnapDrive on this system, follow the instructions in the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
4	Install SnapManager for SQL Server according to the instructions in Chapter 3, "Installing or Upgrading SnapManager," on page 35.

Installing or Upgrading SnapManager 3

The following topics describe the procedures for installing or upgrading to SnapManager 2.1 for SQL Server and then starting the application for the first time. Other sections pertaining to tasks you might need to use at some point after SnapManager has been installed.

Installing SnapManager 2.1 for SQL Server:

If you are installing SnapManager for the first time, go to one of the following sections, depending on your target environment:

- "Installing SnapManager on a stand-alone Windows host system" on page 36
- "Installing SnapManager into an MSCS environment" on page 47

If an earlier version of SnapManager is currently installed, go to the following section:

- "Upgrading to SnapManager 2.1" on page 51

After you have installed SnapManager 2.1, start the application for the first time as described in Chapter 4, "Starting SnapManager for the First Time," on page 67.

After SnapManager 2.1 for SQL Server has been installed:

If you want to remove all components of SnapManager, go to the following section:

- "Uninstalling SnapManager" on page 58

  Unless it is specified for a particular upgrade path or for a particular troubleshooting situation, you do not need to uninstall SnapManager before reinstalling it or upgrading to a newer version.

  Attention

  If you plan to reinstall SnapManager later, be sure to record the drive letter and path of your current SnapInfo directories. After you reinstall SnapManager, be sure to reconfigure SnapManager to use those same SnapInfo directory locations. This saves SnapManager's records of backups taken before you uninstalled and reinstalled SnapManager.

If you need to reinstall SnapManager (to fix missing or corrupt files, shortcuts, and registry entries), go to the following section:

- "Reinstalling SnapManager" on page 65

Installing SnapManager on a stand-alone Windows host system

This section describes how to install SnapManager on a stand-alone Windows host system used for any of the following purposes:

- The production SQL Server computer
- A remote administration server
- A remote verification server

See the following topics for more information:

- "Comparison of the two installation modes" on page 36
- "Interactive installation mode" on page 37
- "Unattended installation mode" on page 42

Note

You do not need to stop SQL Server instances before or during the SnapManager software installation process.

Related topics:

- "Verifying Windows host system requirements" on page 22
- "Installing SnapManager into an MSCS environment" on page 47

Comparison of the two installation modes

The software installation utility for SnapManager can be run in either interactive mode or unattended mode. These two modes are described in the following table.

Feature	SnapManager installation mode	
	Interactive	**Unattended**
Access	Require user interaction and access to the graphical user interface	Allows automated installation by executing a script or command-line command
Minimum required input	◆ Acceptance of the terms of the license agreement ◆ SnapManager service account ❖ User name ❖ Password	

Feature	SnapManager installation mode	
	Interactive	Unattended
Optional input	◆ User name ◆ Organization name ◆ SnapManager server-side license key ◆ SnapManager installation directory	
SnapManager software license agreement	Displayed in the installation utility	Displayed at the command-line if you pass a specific parameter to the installation utility
After the installation finishes	If a system reboot is required to activate new software, a dialog box appears and prompts you to choose whether you want to reboot the target system.	If a system reboot is required to activate new software, a dialog box appears and prompts you to choose whether you want to reboot the target system. You can override this default behavior by including an optional command-line parameter.

Interactive installation mode

To install SnapManager using the software installation utility in interactive mode, complete the following steps.

Note

You can also install SnapManager in unattended mode, as described in "Unattended installation mode" on page 42. For a comparison of the two installation modes, see "Comparison of the two installation modes" on page 36.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">Install the software from CD-ROM or download it.

Attention
Do not use Terminal Services for any type of SnapManager administration, because you might miss critical information that is displayed only in pop-up boxes at the system console.</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You are obtaining the software from the network</td><td>Download the SnapManager package from the network, save it on the Windows host system, then launch the SnapManager installation package by double-clicking it in your Windows Explorer screen.

Result: The software installation utility starts and opens to the Welcome screen.</td></tr>
<tr><td></td><td>You are installing the software from CD-ROM</td><td>Browse to the SnapManager installation package and double-click <code>setup.exe</code>.

Result: The software installation utility starts and opens to the Welcome screen.</td></tr>
<tr><td colspan="3">Welcome</td></tr>
<tr><td>2</td><td colspan="2">In the Welcome screen, click Next.

Result: The License Agreement screen appears.</td></tr>
</table>

Step	Action
License agreement	
3	Read the license agreement. If you accept the terms of the agreement, select the “I accept....” option and then click Next. **Result:** The Customer Information screen appears.
Customer information	
4	In the Customer Information screen, specify the user name, the organization name, and the SnapManager license type. **a.** In the User Name box, enter your user name. **b.** In the Organization box, enter your organization name. **c.** In the License Type option, select your SnapManager license type. If you have a storage-system-side license for SnapManager, select Per Filer, and be sure that the SnapManager system-side license is enabled on the storage system. If you have an SQL-Server-side license for SnapManager, select Per Server. **d.** If you selected the Per Server license type, use the License Key box to enter the license key for your server-side license. If you do not have a license key, you can leave the License Key box empty for now and enter your server-side license key later from the SnapManager interface. To do this, first go to the menu bar and select Help > About to open the About SnapManager dialog box. From that dialog box, click Update License to open the Update Server-Side License dialog box and specify your license key.
5	Click Next. **Result:** The Destination Folder screen appears.

Step	Action
Destination folder	
6	Note the full path of the folder in which SnapManager will be installed. The default installation directory for SnapManager 2.1 for Microsoft SQL Server is as follows: `C:\Program Files\NetApp\SnapManager for SQL Server\` **Note** SnapManager 2.1 for Microsoft SQL Server—like SnapManager 2.0 and 1.5 for Microsoft SQL Server—is installed in a different directory than SnapManager 1.0 for Microsoft SQL Server.
7	Optional. If you want to install SnapManager in a directory other than the default installation directory, do the following: **a.** Click Change to open the Change Current Destination Folder dialog box. **b.** Browse to an alternate installation directory. **c.** Click OK to close the dialog box. The Destination Folder screen displays the new specified installation directory path. **d.** Record the newly specified installation directory path.
8	Click Next. **Result:** The SnapManager Server Identity screen appears.

Step	Action
SnapManager server identity	
9	In the Account box of the SnapManager Server Identity screen, specify the user account you want to use to run SnapManager. This account must meet the following requirements: ◆ The account must have administrator privileges on the SQL Server computer. ◆ If you are running in Windows authentication mode, the account must also have system administrator server privileges. For more information, see "SnapManager service account requirements" on page 26 and "SnapManager service account requirements in workgroup mode" on page 27. If SnapDrive is installed and configured, the text box is prepopulated with the account that SnapDrive is configured for. Otherwise, browse to locate and select the account name. The user account name is specified in either of the following formats: ◆ *DomainName\UserName* ◆ *UserName@DomainName*
10	In the Password box and in the Confirm Password box, enter the user password.
11	Click Next. **Result:** The Ready to Install the Program screen appears. All of the installation specifications are complete.
Ready to install the program	
12	Optional. Review or change your current installation specifications before proceeding by clicking Back.
13	To proceed with the installation using your current specifications, click Install. **Result:** The installation process begins, and the Installing SnapManager for SQL Server screen appears. The screen displays the progress of the installation process.

Step	Action
Installing SnapManager for SQL Server	
14	Wait until the InstallShield Wizard Completed screen appears, then click Finish to exit the software installation utility.
15	Proceed to Chapter 4, "Starting SnapManager for the First Time," on page 67.

Unattended installation mode

This topic describes how to install SnapManager by running the software installation utility from a command-line. This enables you to install the SnapManager software under the control of a script for an unattended installation.

Note

You can also install SnapManager in interactive mode, as described in "Interactive installation mode" on page 37. For a comparison of the two installation modes, see "Comparison of the two installation modes" on page 36.

How to invoke the software installation utility: The command name you specify depends on the SnapManager installation media you will be accessing.

SnapManager installation media	Application name to be run
Software CD	`setup.exe` (located on the CD-ROM)
Software package downloaded from the network	`SMSQL2.1.exe` (the name of the software package itself)

How to display the software license agreement: To display the SnapManager software license agreement, enter the following command either directly at the command-line or through a script:

CommandName `/v"SHOWEULA=Yes /qb"`

CommandName is either `setup.exe` or `SMSQL2.1.exe`, depending on the SnapManager installation media being used.

Note

The SHOWEULA=Yes command argument cannot be used in combination with other command-line arguments.

How to start the unattended installation: To install SnapManager in unattended mode, enter the following command either directly at the command-line or through a script:

CommandName /v"[AGREETOLICENSE={Yes|No}] [USERNAME=*UserName*] [COMPANYNAME=*CompanyName*] [ISX_SERIALNUM=*LicenseKey*] [INSTALLDIR=*InstallationDirectory*] SVCUSERNAME=*Domain\UserName* SVCUSERPASSWORD=*Password* SVCCONFIRMUSERPASSWORD=*PassWord* [REBOOT=0] [/L* *TempDirPath\LogFileName*] /qb"

The following table describes each of the parameters.

Command or parameter	Description
CommandName	Either setup.exe or SMSQL2.1.exe, depending on the SnapManager installation media being used.
AGREETOLICENSE={Yes|No}	Set this parameter to Yes only if you have read and accept the terms of the SnapManager software license agreement. The installation will not proceed unless you specify this parameter with a value of Yes. If not specified, the default value No is used, and the installation will not proceed.
USERNAME=*UserName*	Optional. If not specified, the default value is retrieved from the registry.
COMPANYNAME=*CompanyName*	Optional. If not specified, the default value is retrieved from the registry.
ISX_SERIALNUM=*LicenseKey*	Optional. Only used to specify an SQL server-side license for SnapManager.

Command or parameter	Description
INSTALLDIR=*InstallationDirectory*	Optional. If not specified, the default installation directory is used: `C:\Program Files\NetApp\SnapManager for SQL Server\`
SVCUSERNAME=*Domain\UserName* SVCUSERPASSWORD=*Password* SVCCONFIRMUSERPASSWORD=*Password*	The account from which SnapManager is to be run. See "SnapManager service account requirements" on page 26 and "SnapManager service account requirements in workgroup mode" on page 27.
REBOOT=0	Optional. After the installation finishes, the installation utility automatically reboots the Windows host system if that is required to activate updated software. If you specify this option, however, the system will not be rebooted.

Command or parameter	Description
`/L* TempDirPath\LogFileName`	Optional. If you specify this option, detailed information about the installation will be written to the specified log file. This information can be used to investigate details about how a particular instance of SnapManager for Microsoft SQL Server is installed. The asterisk ("`*`") is a wildcard character specifying that all of the installation information (such as status messages, nonfatal warnings, and error messages) is to be logged. *TempDirPath* is the fully qualified name of the directory in which the installation log will be created or overwritten. *LogFileName* is the name of the file to which the installation information will be written.

Examples: Consider the following two scenarios.

Installation details	Scenario 1	Scenario 2
Installation media to be used	CD inserted into the CD-ROM drive on `E:\`	Installation package downloaded to `C:\NetApp\downloads\`
Service account to be specified	Account user name: `MKTG2\Administrator` Account password: `STeeL`	
SnapManager license	Filer-side	Server-side (license key `ABCDEFGHIJKLMN`)

For scenario 1, you would enter the following at the command-line or from a script:

```
E:\setup.exe /v"AGREETOLICENSE=Yes
SVCUSERNAME=MKTG2\Administrator SVCUSERPASSWORD=STeeL
SVCCONFIRMUSERPASSWORD=STeeL /qb"
```

For scenario 2, you would enter the following at the command-line or from a script:

```
C:\NetApp\downloads\SMSQL2.1.exe /v"AGREETOLICENSE=Yes
SVCUSERNAME=MKTG2\Administrator SVCUSERPASSWORD=STeeL
SVCCONFIRMUSERPASSWORD=STeeL ISX_SERIALNUM=ABCDEFGHIJKLMN /qb"
```

General procedure: To install SnapManager using the software installation utility in unattended mode, complete the following steps.

Step	Action
1	Access the command-line of the target host system. **Note** You do not need to stop SQL Server instances before or during the SnapManager software installation process.
2	At the command-line, enter either the command string or the name of the script that executes the command string. **Result:** The software installation utility installs the SnapManager for Microsoft SQL Server software in the specified directory. If the `INSTALLDIR` parameter is not specified, the utility installs the software in the default directory `C:\Program Files\NetApp\SnapManager for SQL Server\`.
3	Proceed to Chapter 4, "Starting SnapManager for the First Time," on page 67.

Installing SnapManager into an MSCS environment

This section describes how to install SnapManager into an MSCS environment.

See the following topics for more information:

- "System configurations for SnapManager on a Windows cluster" on page 47
- "Disk requirements for SnapManager on a Windows cluster" on page 48
- "Installing SnapManager and creating a new Windows cluster" on page 49

Note

You do not need to stop SQL Server instances before or during the SnapManager software installation process.

Related topics:

- "Verifying Windows host system requirements" on page 22
- "Installing SnapManager on a stand-alone Windows host system" on page 36

System configurations for SnapManager on a Windows cluster

When installing SnapManager on an MSCS cluster, consider the following system configuration requirements and limitations.

SnapManager must be installed on all nodes: SnapManager must be installed on all nodes of the cluster so that SnapManager backup and restore operations can be performed from any node.

Maximum cluster size: SnapManager supports a maximum cluster size of two or four nodes, depending on the operating system.

Operating system	Maximum cluster size
Windows 2000 Advanced Server	Two nodes
Windows Server 2003 Enterprise Edition	Four nodes

Multiple-instance clusters: SnapManager supports multiple-instance clusters, provided that the following additional system requirements are met:

- Each instance must have its own LUNs that cannot be used by other instances.
- Each instance must be created in its own *cluster group*.

- All LUNs assigned to a given instance must be in the cluster group for that instance.

Disk requirements for SnapManager on a Windows cluster

In a clustered environment, SnapManager disk requirements vary, depending on the cluster configuration.

Single-instance cluster example: In an Active/Passive two-node configuration, there are two clustered nodes and one SQL Server instance. If the active node (the node running SQL Server) fails, the cluster transfers the SQL Server instance to the other (previously passive) node, which then becomes the active node and takes over the LUNs previously used by the failed node.

For a single-instance SQL Server cluster, assuming that your SQL Server data is on a shared resource, your disk requirements are the same as for a stand-alone SQL Server system, with the addition of one LUN for the quorum disk. A minimum of three LUNs are required:

- One LUN for the databases
- One LUN for the SnapInfo directory
- One LUN for the quorum disk

Multiple-instance cluster example: In an Active/Active two-node configuration, there are two clustered nodes and an SQL Server instance running on each node. If one node fails, the other node takes over the SQL Server instance running on the failed node. Because both nodes need to be able to run an active SQL Server instance, each node requires its own disks, as if it were a self-contained, stand-alone system. In addition, one extra LUN is needed for the quorum disk. Whether you use a hard disk or a LUN as the quorum disk, each configuration requires a minimum of five disks used for the following purposes:

- For node 1
  - One LUN to store the SQL Server 2000 or 2005 databases
  - One LUN to store the SnapInfo directory
- For node 2
  - One LUN to store the SQL Server 2000 or 2005 databases
  - One LUN to store the SnapInfo directory
  - One LUN or hard disk to be used as the quorum disk

Each node must be able to own all clustered disk resources in a cluster at any given time.

For more information about MSCS clustering with SQL Server, see the SQL Server 2000 Failover Clustering document at the following URL: http://www.microsoft.com/sql/techinfo/administration/2000.

Installing SnapManager and creating a new Windows cluster

The following procedure assumes the following:

- You have two Windows host systems that you are ready to configure into an MSCS cluster.
- Microsoft SQL Server 2000 or 2005 is not yet installed on either system.
- You will be using a LUN as the quorum disk.

To create the cluster and then install and configure SnapManager on the cluster, complete the following steps.

Step	Action
1	Install one of the following operating systems on all nodes of the cluster: ◆ Windows 2000 Advanced Server with SP4 ◆ Windows Server 2003 Enterprise Edition See your Microsoft documentation for information on how to do this.
2	Install and configure SnapDrive and the MSCS cluster using the detailed instructions in the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
3	Create a quorum disk. See the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
4	Configure MSCS. See both the *SnapDrive Installation and Administration Guide* (for the version of SnapDrive that you are running) and your Microsoft documentation.
5	Create the shared disks that will be used as the physical disk resources for the SQL Server instance. See both the *SnapDrive Installation and Administration Guide* (for the version of SnapDrive that you are running) and your Microsoft documentation.

Step	Action
6	Install SQL Virtual Server on the MSCS cluster. See your SQL Server documentation for information on how to do this.
7	If you are using SnapDrive 4.1, set the registry variable so that SnapDrive can restore the MSCS LUN without taking the LUN resource offline. **Note** This must be done on all nodes of the cluster. For detailed instructions, see the NetApp Knowledge Base article KB10590 "Restoring a Clustered LUN Resource without Taking the LUN Resource Offline" on the NOW site.
8	Install and configure SnapManager for SQL Server *on all nodes of the cluster.* To do this, go to "Installing SnapManager on a stand-alone Windows host system" on page 36, and use the procedures described in "Interactive installation mode" on page 37. **Note** You do not need to stop SQL Server instances before or during the SnapManager software upgrade process.
9	Start SnapManager (as described in Chapter 4, "Starting SnapManager for the First Time," on page 67).
10	Run the Configuration Wizard (as described in Chapter 7, "Using the SnapManager Configuration Wizard," on page 101).
11	Perform a test move group across the nodes to verify that you have your MSCS environment correctly configured.

Note

If an MSCS move group occurs during a SnapManager operation (for example, if the node that owns the resources fails), you need to restart the SnapManager user interface and run SnapManager Backup.

Upgrading to SnapManager 2.1

This section contains procedures for upgrading to SnapManager 2.1 for Microsoft SQL Server on a Windows host system that is running SnapManager 2.0 or 1.5. See the following topics for more information:

- "Pre-upgrade checklist" on page 51
- "Comparison of the two upgrade modes" on page 52
- "Interactive upgrade mode" on page 52
- "Unattended upgrade mode" on page 55

Note

You do not need to stop SQL Server instances before or during the SnapManager software upgrade process.

Pre-upgrade checklist

Before you upgrade from SnapManager 1.0, 1.5, 2.0 to SnapManager 2.1for Microsoft SQL Server, be sure that your Windows host systems and the storage system are running the supported software versions as described in Chapter 2, "Preparing to Install or Upgrade SnapManager," on page 19.

The following paragraphs summarize these software upgrade considerations.

Microsoft SQL Server 2000 or 2005: Be sure that the appropriate Service Pack is installed for the Microsoft SQL Server that you are running. This is described in "Verifying Windows host system requirements" on page 22.

SnapDrive: SnapManager 2.1 for Microsoft SQL Server is supported by SnapDrive 4.1, x64 and x86 versions. SnapManager 2.0 for Microsoft SQL Server is supported by SnapDrive 4.0, 3.2 and SnapDrive 3.1.1R1, whereas SnapManager 1.5 is supported by these versions of SnapDrive as well as older versions, and SnapManager 1.0 is supported by SnapDrive 2.0.1. If SnapDrive 3.2 or 3.1.1R1 is not already installed on your SQL Server computer, you will need to upgrade to one of those versions. For more details, see "Verifying Windows host system requirements" on page 22.

Data ONTAP: If you need to upgrade SnapDrive on your Windows host systems, be sure that your storage system is running a compatible version of Data ONTAP, as described in "Verifying storage system requirements" on page 31.

For the latest information: For the most current information about SnapManager and SnapDrive, see "Current information" on page 21. The listed documents (and all other NetApp documents) are available at the NOW site at http://now.netapp.com/NOW/knowledge/docs.

Comparison of the two upgrade modes

The software installation utility for SnapManager can be run in either *interactive* mode or *unattended* mode. These two modes are described in the following table.

Comparison	SnapManager upgrade mode	
	Interactive	**Unattended**
Access	Requires user interaction and access to the graphical user interface	Allows automated upgrade by executing a script or command-line command
SnapManager software license agreement	Displayed in the software installation utility.	Displayed at the command-line if you pass a specific parameter to the installation utility

Interactive upgrade mode

To upgrade SnapManager using the software installation utility in interactive mode, complete the following steps.

Note

You can also upgrade SnapManager in unattended mode, as described in "Unattended upgrade mode" on page 55. For a comparison of the two upgrade modes, see "Comparison of the two upgrade modes" on page 52.

Step	Action
1	Exit SnapManager, if you have not already done so. **Note** You do not need to stop SQL Server instances before or during the SnapManager software upgrade process.

Step	Action
2	If you have not already done so, back up system resources using an industry-standard backup utility. **a.** Back up the operating system, including the system state, the boot and system drives, and the registry. **b.** Back up your SQL Server databases and transaction log files. **c.** Use the NTBackup utility that is part of Windows 2000 Server to create and maintain a current emergency repair disk (ERD). In Windows Server 2003, a repair disk is created as part of running the ASR (Automatic System Recovery) Wizard.
3	If you have not already done so, verify that your host system meets the minimum requirements. **Note** If the system is currently running SnapManager 1.5 for Microsoft SQL Server, you will need to upgrade your version of SnapDrive. For details, see "Verifying Windows host system requirements" on page 22.
4	If you have not already done so, verify that your storage system meets the minimum requirements. **Note** If the Windows host system is currently running SnapManager 1.5 for Microsoft SQL Server, you will need to upgrade your version of Data ONTAP on the storage system to comply with the newer version of SnapDrive on the Windows host system. For details, see "Verifying storage system requirements" on page 31.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>5</td><td colspan="2">Start the software installation utility for SnapManager from either the SnapManager CD-ROM or from the network.

Attention
Do not use Terminal Services for any type of SnapManager administration, because you might miss critical information that is displayed only in pop-up boxes at the system console.</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You are obtaining the software from the network</td><td>Download the SnapManager package from the network, save it on the SQL Server, then launch the SnapManager installation package by double-clicking it in your Windows Explorer screen.

Result: The software installation utility starts and opens to the Welcome screen.</td></tr>
<tr><td></td><td>You are installing the software from CD-ROM</td><td>Browse to the SnapManager installation package and double-click <code>setup.exe</code>.

Result: The software installation utility starts and opens to the Welcome screen.</td></tr>
<tr><td>6</td><td colspan="2">Click Next.

Result: The Program Maintenance screen appears.</td></tr>
<tr><td>7</td><td colspan="2">In the Program Maintenance screen, leave the Repair/Upgrade option selected and then click Next.

Result: The Ready to Install screen appears.</td></tr>
<tr><td>8</td><td colspan="2">In the Ready to Install screen, click Install.

Result: The installation begins. When the installation is completed, the InstallShield Wizard Completed screen appears.</td></tr>
</table>

Step	Action
9	In the InstallShield Wizard Completed screen, click Finish to exit the software installation utility.
10	Proceed to Chapter 4, "Starting SnapManager for the First Time," on page 67.

Unattended upgrade mode

This topic describes how to upgrade SnapManager by running the software installation utility from a command-line. This enables you to upgrade the SnapManager software under the control of a script for an unattended upgrade.

Note

You can also install SnapManager in interactive mode, as described in "Interactive upgrade mode" on page 52. For a comparison of the two installation modes, see "Comparison of the two upgrade modes" on page 52.

How to invoke the software installation utility: The command name you specify depends on the SnapManager installation media you will be accessing.

SnapManager installation media	Application name to be run
Software CD	`setup.exe` (located on the CD-ROM)
Software package downloaded from the network	`SMSQL2.1.exe` (the name of the software package itself)

How to start the unattended installation: To install SnapManager in unattended mode, enter the following command either directly at the command-line or through a script:

CommandName /v"REINSTALLMODE=vomus REINSTALL=ALL [/L* *TempDirPath\LogFileName*] /qb"

The following table describes each of the parameters.

Command or parameter	Description
CommandName	Either `setup.exe` or `SMSQL2.1.exe`, depending on the SnapManager installation media being used.
`REINSTALLMODE=vomus`	Causes the software installation utility to recache the new software package.
`REINSTALL=ALL`	Causes the software installation utility to update all the software components.
/L TempDirPath\LogFileName*	Optional. If you specify this option, detailed information about the installation will be written to the specified log file. This information can be used to investigate details about how a particular instance of SnapManager for Microsoft SQL Server is installed. The asterisk ("*") is a wildcard character specifying that all of the installation information (such as status messages, nonfatal warnings, and error messages) should be logged. *TempDirPath* is the fully qualified name of the directory in which the installation log will be created or overwritten. *LogFileName* is the name of the file to which the installation information will be written.

Examples: Consider the following two scenarios.

Installation details	Scenario 1	Scenario 2
Installation media to be used	CD inserted into the CD-ROM drive on `E:\`	Installation package downloaded to `C:\NetApp\downloads\`
Installed version of SnapManager	NetApp SnapManager 2.0 or 2.1 for Microsoft SQL Server	

For scenario 1, enter the following at the command-line:

```
E:\setup.exe /v"REINSTALLMODE=vomus REINSTALL=ALL /qb"
```

For scenario 2, enter the following at the command-line:

```
C:\NetApp\downloads\SMSQL2.1.exe /v"REINSTALLMODE=vomus
REINSTALL=ALL /qb"
```

General procedure: To install SnapManager using the software installation utility in unattended mode, complete the following steps.

Step	Action
1	Access the command-line of the target host system.
2	Exit SnapManager, if you have not already done so. **Note** You do not need to stop SQL Server instances before or during the SnapManager software upgrade process.
3	At the command-line, either enter the command string or enter the name of the script that executes the command string. **Result:** The software installation utility installs the SnapManager for Microsoft SQL Server software.
4	Proceed to Chapter 4, "Starting SnapManager for the First Time," on page 67.

Uninstalling SnapManager

This section describes how to uninstall the SnapManager software. See the following topics for more information:

- "Before you uninstall SnapManager" on page 58
- "Comparison of the two uninstallation modes" on page 59
- "Uninstalling SnapManager in interactive mode" on page 60
- "Uninstalling SnapManager in unattended mode" on page 61

Note

Unless it is specified for a particular upgrade path or for a particular troubleshooting situation, you do not need to uninstall SnapManager before reinstalling it or upgrading to a newer version.

Related topics:

- "Installing SnapManager on a stand-alone Windows host system" on page 36
- "Installing SnapManager into an MSCS environment" on page 47
- "Upgrading to SnapManager 2.1" on page 51.
- "Reinstalling SnapManager" on page 65

Before you uninstall SnapManager

If you have used SnapManager to manage your SQL Server databases and *you plan to reinstall* SnapManager later, be sure to *record the drive letter and path of the SnapInfo directory locations before you uninstall* SnapManager.

Single SnapInfo directory: If you have set up a single SnapInfo directory for all databases on this host, then record the drive letter and path of the LUN that contains a single SnapInfo directory for all SQL Server instances and their associated databases.

Advanced SnapInfo directories: If you have set up multiple SnapInfo directories, then record the following information:

- The drive letter and path of the LUN that contains the default SnapInfo directory for all SQL Server instances
- The drive letter and path of any LUN that contains the default SnapInfo directory for one or two LUNs shared by multiple databases (if configured)
- The drive letter and path of any LUN that contains the SnapInfo directory for an individual database (if configured)

SnapManager Reports records the current SnapInfo directory locations in the most recent logs contained in the *Backup folder* and in the *Config folder*.

After you reinstall SnapManager, be sure to reconfigure SnapManager with *the same* SnapInfo directory locations that were used by SnapManager previously.

Attention

If you configure SnapManager with *different* SnapInfo directory locations than used previously, then SnapManager will no longer have records of any backups taken before the reinstallation of SnapManager occurred. As a result, your prior backup sets could be invalidated or deleted the next time you perform a backup operation.

Comparison of the two uninstallation modes

The software installation utility for SnapManager can be run in either interactive mode or unattended mode. These two modes are described in the following table.

Comparison	SnapManager uninstallation mode	
	Interactive	**Unattended**
Access	Require user interaction and access to the graphical user interface	Allows automated uninstallation by executing a script or command-line command
Tool used	The Windows utility Add or Remove Programs (in the Control Panel)	The software installation utility for SnapManager for Microsoft SQL Server
Options	You can also remove the Report directory.	

Uninstalling SnapManager in interactive mode

To uninstall SnapManager for Microsoft SQL Server and all its components by using the Windows Add or Remove Programs utility, complete the following steps.

Note

You can also uninstall SnapManager in unattended mode, as described in "Uninstalling SnapManager in unattended mode" on page 61. For a comparison of the two uninstallation modes, see "Comparison of the two uninstallation modes" on page 59.

Step	Action	
1	If SnapManager is running, close it. **Note** You do not need to stop SQL Server or remove the SQL Server databases from the LUNs before you uninstall SnapManager. SQL Server will continue to run during the uninstallation process and after the uninstallation is completed.	
2	In the Windows 2000 Server or Windows 2003 Control Panel, select Add or Remove Programs, then select the entry for SnapManager for Microsoft SQL Server. **Result:** Both a Change button and a Remove button appear in that entry.	
3	Click one of the buttons, depending to what you want to remove.	
	If...	**Then...**
	You only want to remove the SnapManager software and leave the Report folder	Click Remove.
	You want to remove both the SnapManager software and the Report folder	Do not click Remove. Instead, do the following: **a.** Click Change. **b.** Select the Remove Reports option. **c.** Click Remove.

Step	Action
4	At the prompt, click Yes to proceed with removing the SnapManager software.
5	After the utility finishes removing SnapManager from your computer, close the Add or Remove Programs utility and then close the Control Panel.

Note

In a cluster configuration, be sure to uninstall SnapManager from all nodes of the cluster.

Uninstalling SnapManager in unattended mode

This topic describes how to uninstall SnapManager using the software installation utility in unattended mode. This enables you to uninstall SnapManager under the control of a script for an unattended uninstallation.

Note

You can also uninstall SnapManager in unattended mode, as described in "Uninstalling SnapManager in interactive mode" on page 60. For a comparison of the two uninstallation modes, see "Comparison of the two uninstallation modes" on page 59.

How to invoke the software installation utility: The command name you specify depends on the SnapManager installation media you will be accessing.

SnapManager installation media	Application name to be run
Software CD	`setup.exe` (located on the CD-ROM)
Software package downloaded from the network	`SMSQL2.1.exe` (the name of the software package itself)

What to specify in the command-line: To install SnapManager in unattended mode, enter the following command either directly at the command-line or through a script:

CommandName /v"REMOVE=ALL [REMOVEREPORTFOLDER=1] [/L* *TempDirPath\LogFileName*] /qb"

The following table describes each of the parameters.

Command or parameter	Description
CommandName	Either `setup.exe` or `SMSQL2.1.exe`, depending on the SnapManager installation media being used.
`REMOVE=ALL`	Causes the software installation utility to remove SnapManager (as if you selected the Remove option in the Program Maintenance screen).
`REMOVEREPORTFOLDER=1`	Optional. Causes the software installation utility to remove the Report folder (as if you selected the Remove Report Folder option in the Remove the Program screen).

Command or parameter	Description
/L* TempDirPath\LogFileName	Optional. If you specify this option, detailed information about the installation will be written to the specified log file. This information can be used to investigate details about how a particular instance of SnapManager for Microsoft SQL Server is installed. The asterisk ("*") is a wildcard character specifying that all of the installation information (such as status messages, nonfatal warnings, and error messages) should be logged. *TempDirPath* is the fully qualified name of the directory in which the installation log will be created or overwritten. *LogFileName* is the name of the file to which the installation information will be written.

Examples: Consider the following two scenarios.

Installation details	Scenario 1	Scenario 2
Installation media to be used	CD inserted into the CD-ROM drive on `E:\`	Installation package downloaded to `C:\NetApp\downloads\`
SnapManager license	Filer-side	Server-side (license key `ABCDEFGHJIJKLMN`)

For scenario 1, enter the following at the command-line:

```
E:\setup.exe /v"REMOVE=ALL /qb"
```

For scenario 2, enter the following at the command-line:

```
C:\NetApp\downloads\SMSQL2.1.exe /v"REMOVE=ALL
REMOVEREPORTFOLDER=1 /qb"
```

General procedure: To install SnapManager using the software installation utility in unattended mode, complete the following steps.

Step	Action
1	Access the command-line of the target host system.
2	At the command-line, either enter the command string or enter the name of the script that executes the command string. **Result:** The software installation utility upgrades the SnapManager for Microsoft SQL Server software.
3	Proceed to Chapter 4, "Starting SnapManager for the First Time," on page 67.

Note

In a cluster configuration, be sure to uninstall SnapManager from all nodes of the cluster.

Reinstalling SnapManager

You can reinstall the same version of SnapManager on a Windows host system. This option fixes missing or corrupt files, shortcuts, and registry entries.

Note
You do not need to stop SQL Server instances before or during the SnapManager software reinstallation process.

Unless it is specified for a particular upgrade path or for a particular troubleshooting situation, you do not need to uninstall SnapManager before reinstalling it or upgrading to a newer version.

If you uninstalled SnapManager: If you uninstalled SnapManager, then the reinstallation procedure is identical to a new installation of the software. For installation instructions, see the following topics:

- "Installing SnapManager on a stand-alone Windows host system" on page 36
- "Installing SnapManager into an MSCS environment" on page 47

If you had used SnapManager to manage your SQL Server databases before you uninstalled the SnapManager application, then be sure to configure SnapManager with **the same** SnapInfo directory location or locations that were used by SnapManager prior to the reinstallation.

Attention
If you configure SnapManager with *different* SnapInfo directory locations than used previously, then SnapManager will no longer have records of any backups taken before the reinstallation of SnapManager occurred.

For more information, see "Uninstalling SnapManager" on page 58.

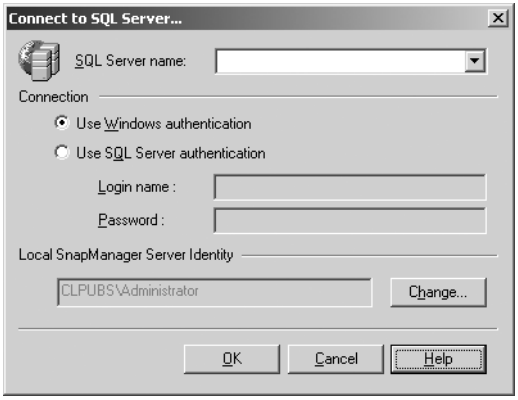
If you did not uninstall SnapManager: See "Upgrading to SnapManager 2.1" on page 51.

After you have installed or upgraded to SnapManager 2.1, start the application for the first time, as described in this section.

Attention

You must run SnapManager from the system console, not from a Terminal Services client. Do not use Terminal Services for any type of SnapManager administration, because you might miss critical information that is displayed only in pop-up boxes at the system console.

To start SnapManager for the first time, complete the following steps.

Step	Action
1	Start SnapManager 2.1 for Microsoft SQL Server. To do this, go to the Start menu and select Program Files > SnapManager for SQL Server. If you have just upgraded to SnapManager 2.1 for Microsoft SQL Server, SnapManager automatically connects to the default SQL Server already configured. Go to Step 4. If you have just installed SnapManager for the first time, the Connect to SQL Server dialog box appears so that you can specify the default SQL Server. 

Step	Action
Specify the default SQL Server.	
2	Select the SQL Server you want to manage. **a.** In the SQL Server box, specify the SQL Server you want to manage. **b.** In the Connection panel, choose the security authentication method to be used to connect to the SQL Server. **c.** If you selected SQL Server authentication, you must also specify the login name and password. **d.** Click OK. For more detailed information, see "Connecting to an SQL Server instance" on page 366. **Result:** SnapManager connects to the specified SQL Server, using the specified security authentication method.
SnapManager creates a share for remote access to the Report directory.	
3	The first time SnapManager is launched, it creates a Report directory share, `SMSQLReportFolder`, to allow for remote administration. The following accounts are granted *Full Control* on this share: ◆ The logged-in user on the local machine ◆ The `Administrators` group on the local machine If another user is logged onto a *remote administration server* with an account that belongs to the `Administrators` group on this machine, that user can connect to this SQL Server and view the SnapManager reports. If SnapManager was just upgraded from SnapManager 2.0, there might be other existing accounts with permissions on the share. ◆ If only the existing group `Everyone` has permissions on the Report directory share, SnapManager removes the Full Control permissions to the Reports directory for that group. ◆ If multiple existing accounts (including `Everyone`) have permissions on the Report directory share, SnapManager does not modify those accounts.

Step	Action
SnapManager checks the SQL Server version.	
4	If SnapManager detects that the SQL Server 2000 or 2005 software does not have the minimum service pack level required by SnapManager, an error message box appears. If this occurs, see the table of "Windows host system requirements" on page 23 and note the requirements for *Microsoft SQL Server*. ◆ If you have a different SQL Server instance that has the minimum service pack required by SnapManager, you can connect to that instance instead. **a.** Open the Connect to SQL Server dialog box. To do this, select Action > Connect to Another SQL Server. **b.** Return to Step 2. ◆ Otherwise, close the SnapManager application, upgrade Microsoft SQL Server as needed, then restart SnapManager.
If SQL Server 2005 is installed, SnapManager checks the MDAC version.	
5	If SnapManager detects that SQL Server 2005 is installed on a Windows host that does not have the MDAC version required by SQL Server 2005 for the installed version of Windows, an event is posted to the Windows Application event log. If this occurs, do the following: **a.** See the table of "Windows host system requirements" on page 23 and note the *operating system* requirements with respect to SQL Server 2005. **b.** Close the SnapManager application, upgrade the MDAC as needed, then restart SnapManager.

Step	Action
SnapManager checks the SnapDrive version.	
6	If SnapManager detects that SnapDrive is installed but that it is a version not supported with this version of SnapManager, a warning dialog box appears. If this occurs, see the "Windows host system requirements" on page 23 and note the requirements for Microsoft SQL Server 2000 or 2005. Close the SnapManager application, upgrade SnapDrive as needed, then restart SnapManager.
SnapManager reminds you to run the Configuration Wizard.	
7	If, after connecting to the default SQL Server, SnapManager detects that no databases have been configured for use with SnapManager, a message box appears and prompts you to run the Configuration Wizard.
8	Click OK. **Result:** The message box closes, and SnapManager automatically launches the Configuration Wizard. 

Step	Action
Plan your database migration before you run the Configuration Wizard.	
9	For now, click Cancel to exit the Configuration Wizard without specifying any database or LUN configuration information yet. After you have planned your database migration (as described in Chapter 6, "Preparing to Migrate SQL Server Databases," on page 85), start the Configuration Wizard manually and specify database and LUN configuration details (as described in Chapter 7, "Using the SnapManager Configuration Wizard," on page 101).

If you are running SnapManager on a Windows Server 2003 machine with the Windows XP theme enabled, SnapManager is displayed with the Windows XP theme also.

Understanding Volume Mount Point 5

Volume mount point introduced in SnapManager 2.1 helps you to surpass the 26-drive letter limitation.

The following topics are covered:

- "Understanding NTFS volume mount point" on page 74
- "SQL Server configuration rules with SnapManager" on page 86
- "Understanding SnapManager support for volume mount points" on page 76
- "Backup and recovery using volume mount point" on page 82
- "SQL Server configurations supported with SnapManager" on page 89

Understanding NTFS volume mount point

This topic contains highlights about NTFS volume mount point:

- "About NTFS volume mount points" on page 74
- "Volume mount point limitations" on page 74
- "Volume mount point limitations in a clustered environment" on page 74

About NTFS volume mount points

A *volume mount point* is a drive or volume in Windows that is mounted to a folder that uses the NTFS file system. A mounted drive is assigned a drive path instead of a drive letter. Volume mount points enable you to exceed the 26-drive-letter limitation. By using volume mount points, you can graft, or mount, a target partition onto a folder on another physical disk.

Volume mount point limitations

You can create volume mount points on either a shared or a dedicated disk. Volume mount points are not supported in the following scenarios:

- When they are used with Windows clusters that are running Microsoft Windows 2000 or earlier.

  For more details, refer to the Microsoft Knowledge Base article 819546.
- When you create a volume mount point on a server cluster, consider the following key items regarding volume mount points:
  - Volume mount points cannot go between clustered and nonclustered disks.
  - You cannot create mount points that refer to the quorum disk.

Volume mount point limitations in a clustered environment

When creating mount points on a server cluster, you must keep these additional limitations in mind:

- Volume mount points are not supported on clustered Windows 2000 Servers.
- The mounted volume must be of the same type as its root:
  - If the root volume is a shared cluster resource, the mounted volume must also be shared.
  - If the root volume is dedicated, the mounted volume must also be dedicated.
- You cannot create mount points to the quorum disk.

- If you have a mount point from one shared cluster resource disk to another, ensure that the disks are in the same group and that the mounted disk resource is dependent on the root disk source.

  For more details, refer to the Microsoft TechNet article 280297.

Understanding SnapManager support for volume mount points

This topic describes SnapManager support for volume mount points, which is new in SnapManager 2.1:

- "Drive letter limitations and individual database restoration" on page 77
- "Mounted volume environments supported by SnapManager" on page 78
- "Mounted volume restrictions with SnapManager" on page 78
- "Mounted volume naming conventions with SnapManager" on page 79
- "Naming conventions for multiple databases sharing same or two LUNs" on page 79
- "Using mounted volumes in SnapManager" on page 79

Related topics:

- "Understanding NTFS volume mount point" on page 74

Drive letter limitations and individual database restoration

Windows supports up to 26 drive letters. For SnapManager to migrate, back up, and restore SQL Server databases, SnapManager requires a minimum of two LUNs to hold SQL Server data, transaction log files, and the SnapInfo directory. You can allocate a maximum of 25 drive letters. Additionally, certain SnapManager operations require more drive letters for the following:

- Databases that need to be restored individually, which require their own LUN assigned an available drive letter in the Windows host. If you have a large number of databases to be individually restored, you might run out of drive letters.
- Performing a remote verification of more than one backup set, which requires a second LUN and therefore another available drive letter or mount point.

With SnapManager 2.1 for SQL, your configuration is not limited to the 26 drive letters supported by Windows. By using the NTFS volume mount point support that is part of SnapDrive, SnapManager can manage SQL databases that are stored on mounted volumes as well as those stored on standard Windows volumes.

Mounted volume environments supported by SnapManager

The following table summarizes the environments in which SnapManager 2.1for SQL supports volumes mounted on Data ONTAP LUNs. Additional details regarding limitations and enforcement imposed by SnapManager are described in subsequent sections of this document.

	Windows host environment			
	Windows 2000 Server		Windows Server 2003	
Microsoft SQL	Stand-alone	Clustered [1]	Stand-alone	Clustered
SQL 2000	Yes	No	Yes	No
SQL 2005	Yes	No	Yes	Yes

[1] Windows 2000 Server does not support mounted volumes in a cluster

Mounted volume restrictions with SnapManager

SnapManager imposes the following restrictions.

No SQL database files on a mount point root: An NTFS volume that hosts mount points cannot store SQL database files. However, transaction log files can reside on such a volume.

Following are the mounted volume restrictions with SnapManager:

- SnapManager Configuration Wizard does not allow you to place SQL database files on an NTFS volume that has mount points.
- SnapManager Backup does not allow you to back up a database on an NTFS volume that has mount points.

Mounted volume naming conventions with SnapManager

A mounted volume is a volume that is mapped to a folder and referenced by a mount point that is created in an empty directory. The SnapManager user interface represents a mounted volume based on the path of a volume mount point that references that mounted volume, as follows:

DriveLetter:\MountPointName

- *DriveLetter* represents the drive letter to which the mounted volume is assigned.
- *MountPointName* represents the mount point name assigned to the empty directory used to reference the mounted volume.

Example: `S:\ My_Mount_Point`

Naming conventions for multiple databases sharing same or two LUNs

SnapManager for SQL provides a specific naming convention for multiple databases using either the same or two different LUNs, The SnapManager user interface represents a mounted volume as follows:

VDISK-MountPoint-*n*

n represents the configuration number of a mount point in a specific system. The actual mount point paths will be saved in the Registry Edit.

Example: `VDisk_MountPoint_001`

Using mounted volumes in SnapManager

The path-style representation of a mounted volume can appear in any part of the SnapManager user interface that refers to LUNs accessed by SnapManager:

- Configuration Wizard screens that include a *Disk List selection:*
  - Where to Move Databases
  - Where to Move Transaction Logs
  - Where to Put the SnapInfo Directory

  LUNs that are referenced more than once: If the LUN is configured with multiple references, each such LUN reference is listed with a label that lists any other references to the same LUN.

  For example, suppose the *drive letter* `M:` and the *mount point* `C:\Mnt_Pnt\` reference the same LUN. In this case, the Disk List selection contains two entries for one LUN:

  ```
  LUN M:  <C:\Mnt_Pnt\>
  LUN C:\Mnt_Pnt\ <M>
  ```

Swap LUN using reference mount point: Suppose a database resides in LUN M, create a reference `C:\Mnt_Pnt1\db` to it using the SnapDrive. Use the Configuration Wizard to detach the database from the original location LUN M and attach it to the reference `C:\Mnt_Pnt1\db` without copying or moving the database files. This operation is called LUN swapping.

- Run the SnapManager for SQL Configuration Wizard.

  SMSQL configuration wizard lists all references to the same LUN. In this case, the Disk List selection contains two entries:

  ```
  LUN M:  <C:\Mnt_Pnt1\db\>
  LUN C:\Mnt_Pnt1\db\ <M>
  ```

- Highlight the database on LUN M and click *Reconfigure.*
- Select `LUN C:\Mnt_Pnt\ <M>` and associate it with the database.
- Press Next to proceed and complete the Configuration Wizard.

  The database is now attached to the `C:\Mnt_Pnt1\db` instead of M.

LUNs that have mounted volumes: If SnapManager accesses a LUN with a volume that is referenced by a mount point, that LUN is listed with a label that indicates this.

For example, suppose the drive letter `J:` references a LUN that hosts a mount point. In this case, the Disk List selection lists that LUN as follows:

```
LUN J:  (MPRoot)
```

The Configuration Wizard does not allow you to store SQL database files on LUNs that host NTFS volume mount points.

- The location of the stored database files is listed in the left navigation pane of the Backup and Verification tab and the Backup wizard screen:
  - Transaction log file path
  - SnapInfo directory path
- The location of the stored database files is listed in the left navigation pane of the SnapManager Restore tab and SnapManager Restore wizard screen:
  - Transaction log file path
  - SnapInfo directory path
- To specify which method is to be used to *access database backup Snapshot copies during database integrity verification*, use the Access LUN in Snapshot tab to assign either a drive letter or select a mount point directory path to access the backup Snapshot copy as a mounted LUN.

  You can access this setting from the following locations within the SnapManager user interface:
  - Configuration Wizard
  - Options menu

- Backup Wizard
- Restore Wizard

For more information, see Appendix C, “Using the Access LUN in Snapshot tab,” on page 376.

Backup and recovery using volume mount point

The following topics are covered:

- "Perform backup and recovery using volume mount point" on page 82
- "Change backup management group with mounted volume" on page 82
- "Delete backup data set on mounted volume" on page 83

Perform backup and recovery using volume mount point

To perform backup and recovery using volume mount points, complete the following tasks:

1. Migrate all database files to a volume mount point. For more information about database migration, see the Chapter 7, "Using the SnapManager Configuration Wizard," on page 101.
2. Create a backup of all the databases residing on volume mount point. For more information about SQL database backup, see the chapter Chapter 9, "Backing Up Databases Using SnapManager," on page 149.

   Note

   In SQL Server, the backup transaction logs backups and dump files are saved to a SnapInfo directory residing on a volume mount point or a drive letter.

3. Restore Snapshot copies residing on a mounted volume. For more information about the restore operation, Chapter 10, "Restoring Databases Using SnapManager," on page 213.

Change backup management group with mounted volume

To change the backup data set management group that resides on a mounted volume, complete the following tasks:

1. Use Configuration Wizard to configure databases on the mount point.
2. Use Backup Wizard to make backups of the databases on the mounted volume.
3. Go to the Backup and verification tab and change the backup dataset management group. SnapManager must rename the Snapshot copies mounted on a mount point appropriately.

Delete backup data set on mounted volume

To delete the backup dataset that resides on mounted volume, complete the following the following tasks:

1. Use Configuration Wizard to configure databases on the mount point.
2. Use Backup Wizard to make backups of several databases on the mounted volume.
3. Go to the Delete Backup tab and delete the databases. SnapManager should be able to delete Snapshot copies and backup metadata that resides on the mounted volume.

Preparing to Migrate SQL Server Databases 6

This section describes how to prepare to migrate your SQL Server 2000 or 2005 databases to LUNs so you can manage backup and restore operations using SnapManager. The following topics are covered:

- "SQL Server configuration rules with SnapManager" on page 86
- "SQL Server configurations supported with SnapManager" on page 89
- "Assessing volume size" on page 96

Related topics:

- Chapter 7, "Using the SnapManager Configuration Wizard," on page 101
- Appendix D, "Configuration Wizard Screens and Dialog Boxes," on page 407

SQL Server configuration rules with SnapManager

Before you configure your SQL Server environment or migrate your database and transaction files to LUNs, read and understand the following topics:

- "Maximum configurations supported by SnapManager" on page 86
- "SQL Server database configurations supported" on page 87
- "SQL Server database configuration restrictions enforced" on page 87
- "SQL Server database configurations to avoid" on page 88
- "Recommendations" on page 88

Maximum configurations supported by SnapManager

The following table lists the maximum configuration capacities tested and supported for the SnapManager environment.

Configuration capacity	Maximum
SQL Server instances per SQL Server computer	6
Databases per SQL Server instance	
For one or two SQL Server instances per SQL Server computer	253
For three or more SQL Server instances per SQL Server computer	100
Databases per SQL Server computer	600
Databases per LUN	100
Databases per storage system volume	100
Filegroups per database	10
Filegroups per SQL Server computer	600
Database files per SQL Server computer	1200
Storage system volumes	
That can be used to store a single database	8
That can be used to store LUNs connected to an individual SQL Server computer	12

Although SnapManager does not prevent you from creating configurations that exceed these limits, such configurations are untested and unsupported. NetApp strongly recommends that you do not exceed any of these limits.

SQL Server database configurations supported

This topic describes some basic SQL Server data configurations supported by SnapManager. For more examples of supported configurations, see "SQL Server configurations supported with SnapManager" on page 89.

Distributing the files of a database: You can spread the files belonging to your database across multiple LUNs on different volumes. However, there are restrictions on this configuration, as described on the following topics:

- "SQL Server database configuration restrictions enforced" on page 87
- "SQL Server database configurations to avoid" on page 88

Multiple databases sharing one or two LUNs: Multiple databases and all their associated files and transaction logs can share one or two LUNs provided that all files belonging to those databases are contained on that LUN or LUNs.

- An illustration of multiple databases sharing one LUN is shown in "Multiple databases on one LUN" on page 90.
- An illustration of multiple databases sharing two LUNs is shown in "Multiple databases sharing two LUNs" on page 91.

In this configuration, all databases in the shared LUN are backed up at the same time. However, you can restore any number of databases from the backup set.

SQL Server database configuration restrictions enforced

Before migrating your SQL Server 2000 or 2005 databases to LUNs, note the following restrictions enforced by the SnapManager Configuration Wizard.

All the files belonging to a single database: For any database that you migrate to LUNs for use with SnapManager, *all of the files* (the data files, as well as the transaction log files) must be migrated to LUNs.

Single database on multiple LUNs: The files belonging to an individual database can be spread across two or more LUNs, as long as those LUNs are not used for storing database files belonging to other databases.

No database files on the same LUN as the SnapInfo directory: You cannot migrate a database to the LUN on which the SnapManager SnapInfo directory resides.

No database files or SnapInfo directory on a SAN boot LUN: You cannot place database files, transaction log files, or a SnapInfo directory on a SAN boot LUN (a LUN configured as a boot device for a SAN host).

No user database files on the SQL Server 2005 root LUN: You cannot migrate a user database to a LUN that hosts the SQL Server 2005 root path, data root path, or SQL error log path.

SQL Server database configurations to avoid

If you add more databases or move databases to different LUNs without using the Configuration Wizard, you can create an invalid configuration that could cause SnapManager backup or restore operations to fail.

Note

Always run the Configuration Wizard after adding or moving SQL Server databases. The Configuration Wizard ensures that the SQL Server databases are stored in valid locations so that SnapManager backup and restore operations can be completed successfully.

Before you migrate your SQL Server data files and transaction log files to LUNs, note the following recommendations against certain invalid configurations that could be created outside of the SnapManager Configuration Wizard.

No other files on LUNs used by any database files: Do not manually store any directories or files (including system paging files) on the LUNs used for the SQL Server database files. These LUNs should store only the SQL Server database files (data files and transaction log files) that are placed there by the SnapManager Configuration Wizard.

No other files on the LUN used by the SnapInfo directory: Do not manually store any directories or files on the LUN used by the SnapInfo directory. This LUN should store only the directories or files that are placed there by the SnapManager Configuration Wizard.

Recommendations

If full-text indexing is enabled on an SQL Server 2005 database, NetApp recommends that you store the full-text search index catalog files on the same LUN as the database files.

SQL Server configurations supported with SnapManager

SnapManager databases can be configured on one or more storage systems. The illustrations in this section show the various ways you can place the data of your SQL Server on storage system volumes.

The supported configurations are as follows:

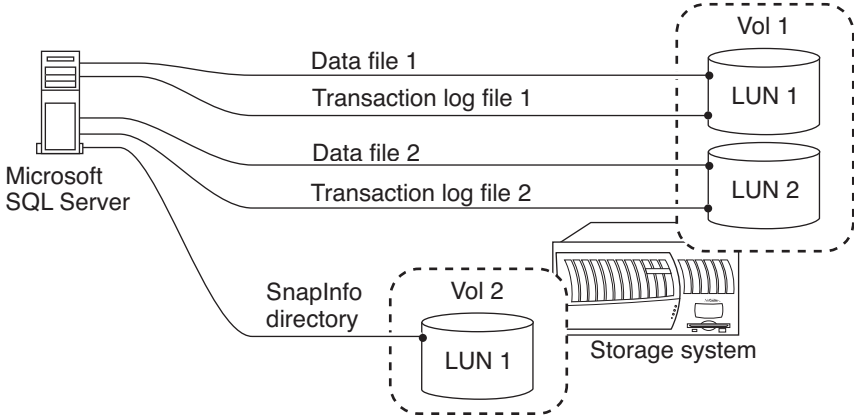
- "Multiple databases on different LUNs within the same volume" on page 90
- "Multiple databases on one LUN" on page 90
- "Multiple databases sharing two LUNs" on page 91
- "Single SQL Server and multiple storage system volumes" on page 92
- "Multiple SQL Servers and one storage system volume" on page 93
- "Multiple SQL Server instances on the same storage system volume" on page 94
- "Multiple SQL Server file groups belonging to the same database on different LUNs" on page 95

Note

If you change the database configuration after performing a SnapManager backup, you might not be able to perform an up-to-the-minute restore using that backup. Therefore, NetApp recommends that you perform a backup immediately following any configuration changes.

Multiple databases on different LUNs within the same volume

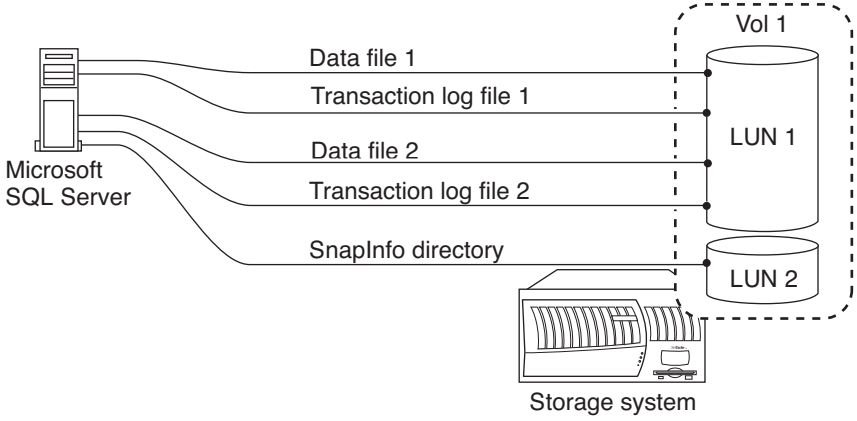
The following supported configuration shows multiple SQL Server databases sharing the same volume but residing on different LUNs. This is the simplest configuration, and it can be applied to an SQL Server that supports between 10 and 20 databases.



Multiple databases on one LUN

The following illustration shows multiple SQL Server databases and all their associated files and transaction logs on one LUN.

When an SQL server configuration has to access many small databases, several databases must share a single LUN. The databases should be grouped according to criteria, such as performance, acceptable downtime, or backup requirements.

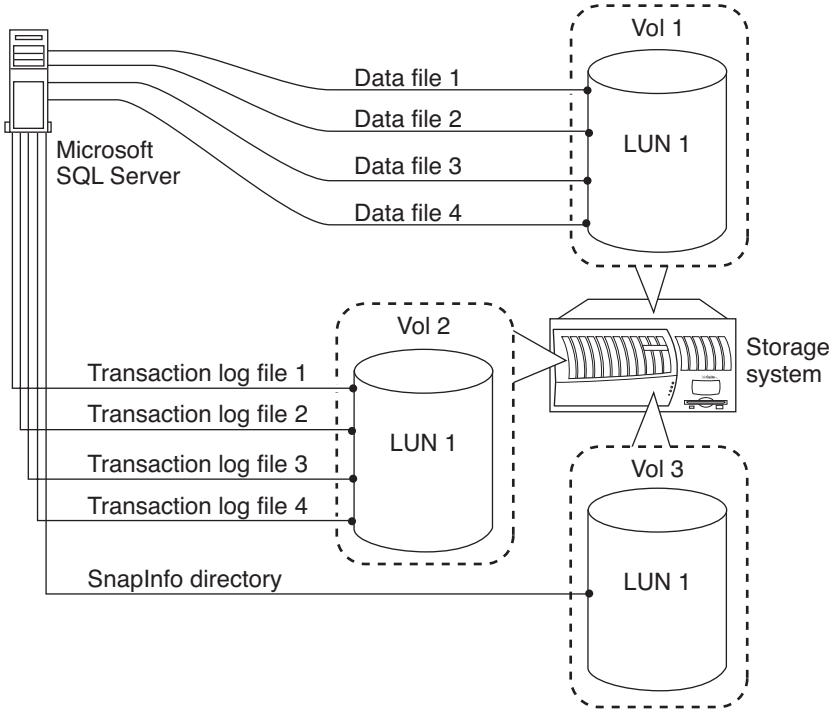


Note

In this configuration, all databases in the shared LUN are backed up at the same time, even if certain databases have not been selected for the backup. However, you have the option to select which databases you want to restore from a multiple-database backup.

Multiple databases sharing two LUNs

The following illustration shows an example of multiple SQL Server databases and all their associated files and transaction logs sharing exactly two LUNs. The database files cannot reside on any other LUNs. This configuration requires a third LUN for the SnapInfo directory. The LUNs can be located on the same or different storage system volumes. The illustration shows an example in which each LUN is located on a different volume.



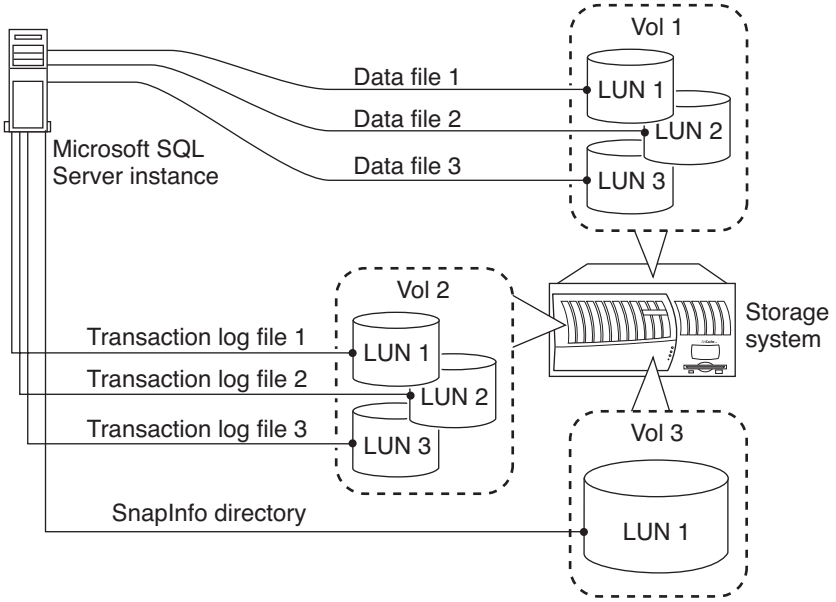
By placing the data files for multiple databases on one LUN and the transaction logs for those databases on the other LUN, SQL database performance is improved by separating the random I/O patterns of the data files from the sequential I/O patterns of the transaction log files.

Note

If you choose to restore only a subset of the databases that reside on one or two LUNs shared by multiple databases, then a stream-based restore method is used rather than the online Snapshot restore method.

Single SQL Server and multiple storage system volumes

The following illustration shows a configuration in which the data and transaction log files of an SQL Server database reside on separate storage system volumes. Placing all transaction logs on one volume and using another volume for all the database files is partly due to performance. If the volume with the data files fails, it is still possible to back up the log file, restore the last full backup, and then apply all backed-up current transaction logs. This configuration requires another volume for the SnapInfo directory.



Multiple SQL Servers and one storage system volume

The following illustration shows multiple SQL Servers with LUNs housing all the server's data files and transaction log files on the same volume.

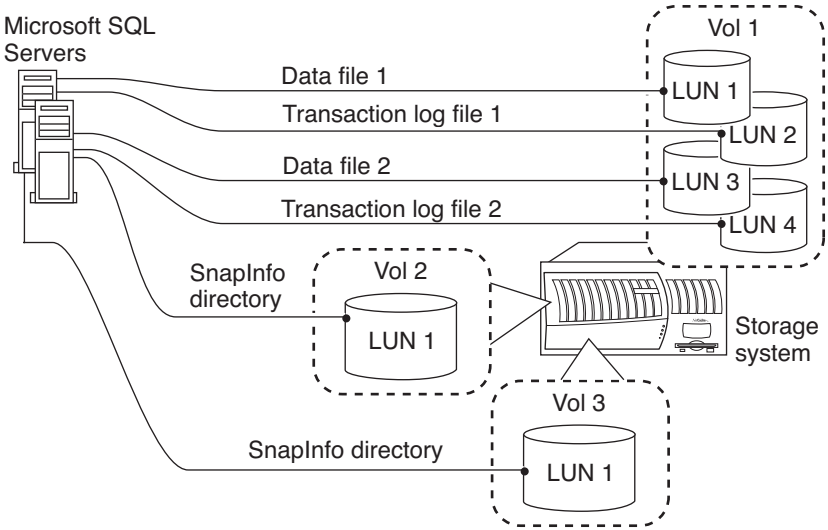
When the SQL Server environment does not generate high I/O load, a single volume can optimize the use of disk and volume space. However, this configuration has two disadvantages:

- If the volume fails, all databases are lost, including the current transaction log files.
- With a single volume housing databases for multiple SQL Servers, there is an increased possibility of creating a busy Snapshot.

  For information about busy Snapshot copies, see "Busy Snapshot error prevents deletion of backup set" on page 210.

Note

If your storage system is equipped with FlexVol™ technology (available beginning with the Data ONTAP® 7.0 release family of software), you can minimize the risk of busy Snapshot copies with this configuration by configuring one FlexVol per SQL Server.

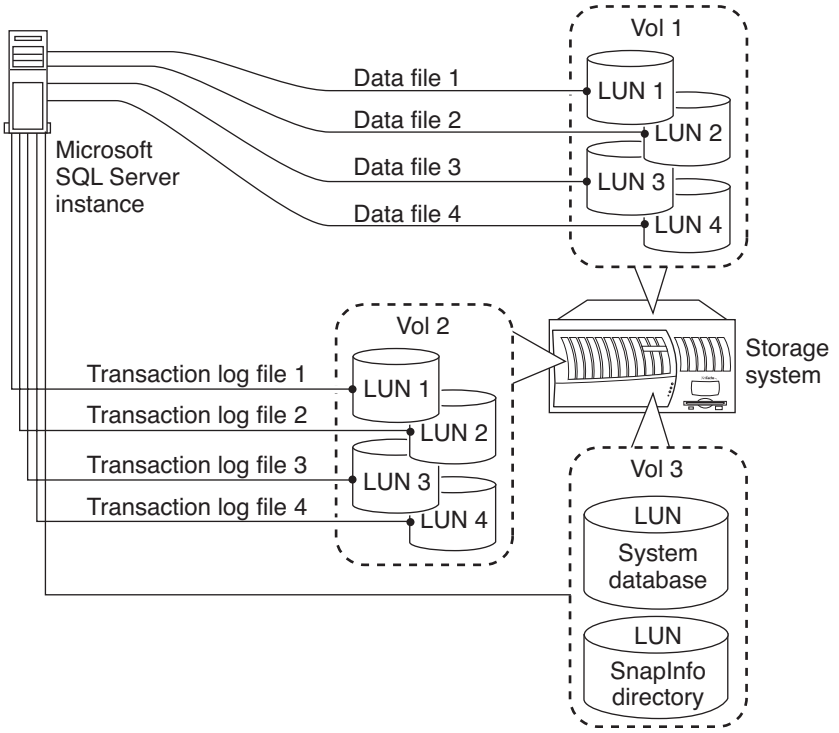


Multiple SQL Server instances on the same storage system volume

The following illustration shows a storage system volume with LUNs containing the data files of multiple SQL Server instances residing on a storage system volume that is different from the volume on which the LUNs for the transaction log files reside.

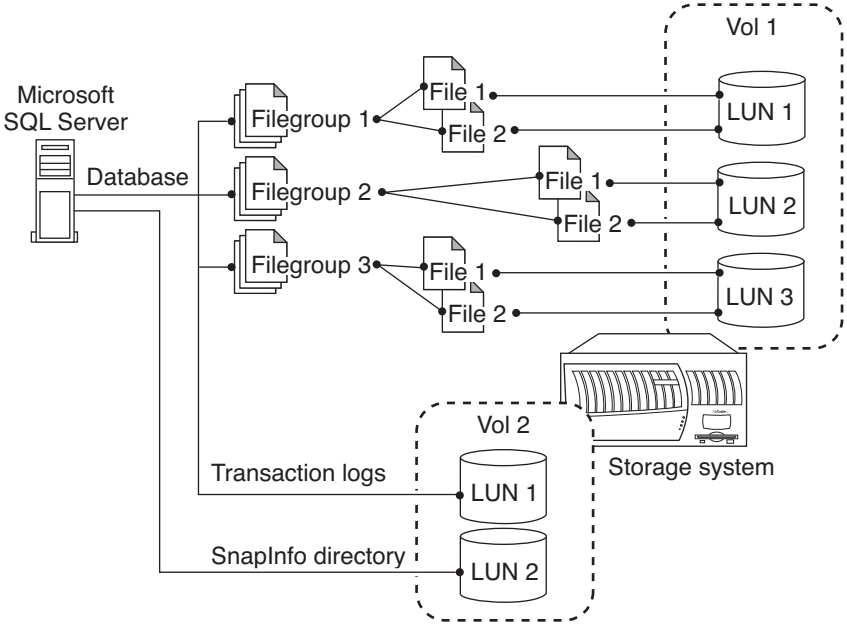
Note

Using three volumes prevents the loss of the complete SQL Server environment and makes it quicker to restore from archive if a volume goes offline.



Multiple SQL Server file groups belonging to the same database on different LUNs

The following illustration shows multiple SQL Server file groups belonging to the same database residing on different LUNs within the same storage system volume.



Assessing volume size

This section describes how to estimate the storage requirements on your storage system.

- "Storage system volume sizing requirements" on page 96
- "LUN size calculations" on page 97
- "Overall storage system volume requirements for a transaction log" on page 98
- "Estimating the amount of space required for a transaction log" on page 98
- "Initial sizing guidelines for new environments" on page 99

For additional details about how to evaluate your space requirements, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

Storage system volume sizing requirements

In addition to the space required for your LUNs, free space on the storage system volume is required to store data that changed between Snapshot copies and the active file system of the LUN. The storage system volume also requires space to store metadata. Allowing for this additional space ensures that making multiple Snapshot copies does not encroach on the LUN objects in the volume.

When you create storage system volumes to hold LUNs, you must take into account the following SnapDrive requirements:

- To create Snapshot copies, the volume storing the LUN must be at least twice the size of that LUN.

  By extension, a storage system volume must have at least twice the capacity of all the LUNs it contains.
- The volume must have enough free space for the number of Snapshot copies you intend to keep online.

  In sizing volumes, take into account how many day's worth of Snapshot copies will be online at the same time.

  The first Snapshot copy lock uses disk space equal to the LUN object size itself (therefore, double the requirement). Additional Snapshot copies increase the amount of required disk space.

LUN size calculations

To ensure that you create volumes to meet these requirements, you must calculate the potential size of your LUNs. The following formulas work best to calculate both the SQL Server database size (LUN size) and the volume size.

Example: Given these requirements, the following parameters can be expressed by the formula shown after them:

- The initial database size is 100 GB.
- The database growth rate is 10 percent per month.
- The change rate of the current database is 15 percent per month.
- The Snapshot copy requirement is four Snapshot copies per day with a total of 12 Snapshot copies (three days' worth of Snapshot copies).
- The default RAID group size is 72 GB x 9 drives.
- The customer only wants to expand the volume every six months.
- Performance is important for the customer, which wants to prevent the volume from being fragmented over time and is willing to allocate 20 percent free space per disk drive.

Calculations:

1. The database size after growth is about 177 GB (rounded up to 180 GB for the LUN size).

   Note

   If the database is growing by 10 percent per month, this does not mean that the database is growing by the same 10 GB every month. Rather, the database grows to about 110 GB in the first month (100 GB x 1.1) and 121 GB in the second month (110 GB x 1.1).

   In general, the formula is as follows:

   <projected future database size>=<present database size> x (<1 + <monthly growth rate percentage in decimal>+<monthly change rate percentage expressed in decimal>)$^{<\text{number of months}>}$

2. Approximately 27 GB of the database changes per month after six months.

3. Minimum space requirements after six months are (180 GB x 2) + (0.15 GB x 12) = 362 GB.

   Note

   You should round up the final space requirements listed in Step 1 to determine the volume size you want to create. For example, round up 368.4 GB to 370 when you are creating your volume size.

4. A 72-GB disk drive has approximately 68 GB of usable volume space, and because 20 percent is allocated for permanent free space, then only 54 GB is usable per disk drive. Therefore, 13 disk drives are needed for data and two disk drives are needed for parity.

Overall storage system volume requirements for a transaction log

The storage system volume requirements for a transaction log require an understanding of the following factors:

- The rate of transactions that modify database tables
- The size of the transactions
- The frequency of the transaction log backup

Note

The key to sizing correctly is to monitor usage over time.

Example: With a table that contains three columns with two indexes defined on column one and column three, for each update operation that adds one data row, there are at least three operations:

1. The actual update to the row (including any old data) is logged.
2. An entry is created for the first index that needs to be updated.
3. An entry is also created for the second index that needs to be updated.

Note

There might be extra entries created if a new index page or data page needs to be created to accommodate the row in the table.

Estimating the amount of space required for a transaction log

The quantity of what is logged is dependent on the underlying table structure and the database activity on the SQL Server.

If the database already exists, then the current transaction log size can be used as is or the transaction log activities can be monitored from the performance monitor with some SQL Server database metrics:

- Log file size (in KB)
- Log file used size (in KB)
- Log bytes flushed per second

Initial sizing guidelines for new environments

If you have set up a new environment, you might want to consider the following initial sizing guidelines and monitor the used size before and after the transaction log is backed up.

Note

The following recommendations are also applicable when you specify the size of the SnapInfo directory.

The recommendations are as follows:

- The transaction log volume size should be 20 percent of the initial database size.
- The minimum transaction log size is 1 MB (default).
- The maximum transaction log size is 100 MB.

Note

The insert, update, and delete functions will increase a transaction log file's size.

Using the SnapManager Configuration Wizard 7

This section describes how to use the SnapManager Configuration Wizard to configure basic SnapManager settings and migrate your SQL Server databases to Data ONTAP LUNs so that they can be backed up and restored using SnapManager. The following topics are covered:

- "How databases are stored on storage system volumes" on page 102
- "Understanding the Configuration Wizard" on page 103
- "Migrating SQL Server databases to LUNs" on page 107
- "Moving multiple SnapInfo directories to a single SnapInfo directory" on page 123
- "Migrating SQL Server databases back to local disks" on page 126

Attention

You must run SnapManager from the system console, not from a Terminal Services client. Do not use Terminal Services for any type of SnapManager administration, because you might miss critical information that is displayed only in pop-up boxes at the system console.

Related topics:

- Chapter 6, "Preparing to Migrate SQL Server Databases," on page 85
- Appendix D, "Configuration Wizard Screens and Dialog Boxes," on page 407

How databases are stored on storage system volumes

During the data migration process, the Configuration Wizard enforces the following rules for storing your SQL Server database files and transaction log files on storage system volumes.

SQL Server database files: Database files that belong to more than one database cannot share a LUN, if all the database files do not reside on that LUN.

SQL Server transaction log files: Transaction logs can reside on the same LUN that stores the data files, or they can reside on another LUN on the same or different volume.

SnapInfo directory: The SnapInfo directory must reside on a LUN that is different from the LUN on which the SQL Server data files and SQL Server transaction logs reside.

Creating a SnapInfo directory: When the Configuration Wizard is used to migrate SQL Server databases from local disk to LUNs, it creates a *SnapInfo directory* that stores SnapManager information about the backup sets and the backed-up transaction logs.

If you use the Configuration Wizard to move databases to LUNs, the wizard performs the following tasks:

1. Detaches the selected databases.
   - Before the Configuration Wizard migrates SQL Server *user databases* to LUNs, SnapManager detaches them. After the migration is complete, the user databases are brought back online.
   - While the Configuration Wizard is migrating SQL Server *system databases* to LUNs, the SQL Server is stopped by SnapManager.

   Note
   Migrating SQL Server databases causes them to be taken offline during the move operation.

2. Moves the SQL Server database files and transaction log files to the correct locations on the specified LUNs.
3. Reattaches the databases.

Understanding the Configuration Wizard

The primary function of the Configuration Wizard is to migrate SQL Server databases to Data ONTAP LUNs so that the databases can be backed up and restored using SnapManager. For more information, go to any of the following topics:

- "What the Configuration Wizard does" on page 103
- "What the Configuration Wizard does not do" on page 103
- "When to use the Configuration Wizard" on page 104
- "Configuration Wizard screens" on page 104
- "Settings configurable only with the Configuration Wizard" on page 106

What the Configuration Wizard does

The Configuration Wizard enables you to move your SQL Server databases in the following ways.

Type of database migration	Purpose
From local disk to LUN	To enable management by SnapManager
From LUN to LUN	If resource management issues require this
From LUN to local disk	If no longer managed using SnapManager

You can choose whether you want to verify your migrated databases and, if you are migrating your databases to LUNs, whether you want to delete your original databases after a successful migration.

The Configuration Wizard also guides you through several application settings. These settings include setting up the SnapInfo directory, enabling notification of SnapManager events using e-mail, and enabling notification of SnapManager events using the storage system Syslog or the AutoSupport feature.

What the Configuration Wizard does not do

Do not use the SnapManager Configuration Wizard to migrate replicated databases or databases used in the replication process. For information about configuring replication-specific databases, see your Microsoft SQL Server documentation.

When to use the Configuration Wizard

You can use the Configuration Wizard in the following situations.

For initial configuration: In order to use SnapManager to back up and restore SQL Server databases, you must use the SnapManager Configuration Wizard to migrate those databases from your SQL servers to the LUNs you configured on your storage system with SnapDrive. The Configuration Wizard also sets up a SnapInfo directory that SnapManager uses to store information about the backup sets and the backed-up transaction logs.

To view or change the database configuration: After the initial configuration, you can rerun the Configuration Wizard at any time to review or make changes to your SQL Server database configuration.

To validate the database configuration: If you add more databases or move databases to different LUNs without using SnapManager, you must run the Configuration Wizard to ensure that the databases are stored in valid locations and to create a mapping between those databases and their respective SnapInfo subdirectories.

Configuration Wizard screens

Brief descriptions of the Configuration Wizard screens are provided in the following paragraphs.

Database Verification Server and Account Authentication: Use this screen to specify the SQL Server you want used to verify the SQL Server databases in your SnapManager backups.

Where to Move Databases: Use this screen to specify the databases (or individual data files or transaction log files) you want to move and where you want them moved.

Number of Databases that Can be Backed Up Concurrently: Use this screen to specify the maximum number of databases that can be backed up concurrently and to choose whether you want to view and optionally change the SQL Server "maximum worker threads" parameter for each SQL Server instance.

Modify SQL Server Max Thread Option: Use this screen to view and optionally change the maximum number of worker threads for each SQL Server instance.

Note

If you choose to change the maximum number of worker threads, the affected SQL Server instances will be rebooted after you complete the Configuration Wizard sequence.

SnapInfo Directory Type: Use this screen to choose how you want the SnapInfo directory or directories configured:

- To specify a single SnapInfo directory for *all* SQL Server instances and their associated databases, select the Single SnapInfo Directory option. After you click Next, the Specify a Single SnapInfo Directory screen appears.
- To specify multiple SnapInfo directories, select the Advanced SnapInfo Directories option. After you click Next, a sequence of screens appear:
  - Where to Put the Default SnapInfo Directory for Each SQL Server Instance

    Use this screen to specify a default SnapInfo directory for all the databases owned by a specific SQL Server instance.
  - Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases

    If you have multiple databases that reside on one or two LUNs, use this screen to specify a common SnapInfo directory location for those databases.
  - Where to Put the SnapInfo Directory for Each Database

    If you want the SnapInfo directory for an individual database to be stored on a LUN other than in the default location for that SQL Server instance, use this screen to specify that information.

Database Migration Options: Use this screen to select options for the database migration:

- Run the DBCC command to verify your migrated databases.
- Delete the original database files after the migration finishes successfully.
- Run UPDATE STATISTICS on tables before detaching the databases.

Add Microsoft iSCSI Service Dependency: Use this screen to specify, on a per-instance basis, whether the Microsoft iSCSI service should be a dependency for the Microsoft SQL Server services.

Configure Automatic E-mail Notification: Use this screen to configure automatic SMTP e-mail notification for event notification.

Configure SnapManager Event Logging to Filer Syslog and AutoSupport: Use this screen to enable Syslog (storage system event log) support and to enable AutoSupport for SnapManager.

Settings configurable only with the Configuration Wizard

The following table lists SnapManager database management settings that can be configured or changed only through the Configuration Wizard. For each setting, the table lists the name of the corresponding screen in the Configuration Wizard.

SnapManager setting	Configuration Wizard screen
Database-LUN mapping	Where to Move Databases
Maximum worker threads for each SQL Server instance	Modify SQL Server Max Worker Thread Option
SnapInfo directory location	SnapInfo Directory Type ◆ Single SnapInfo Directory ◆ Advanced SnapInfo Directories
Migration-specific options: ◆ Run DBCC before migration, after migration, or both? ◆ Delete original databases after successful migration?	Database Migration Options
Microsoft iSCSI Service as a dependency	Add Microsoft iSCSI Service Dependency

Migrating SQL Server databases to LUNs

This section contains procedures for using the Configuration Wizard to migrate SQL Server databases. See the following topics for more information:

- "Prerequisites for migrating databases to LUNs" on page 107
- "Overview of the database migration procedure" on page 108
- "Migrating databases to LUNs using the Configuration Wizard" on page 110
- "Migrating system and resource databases" on page 109

Prerequisites for migrating databases to LUNs

Before you migrate your SQL Server databases, verify that your environment is in the proper state.

- You must use SnapDrive to create the following LUNs:
  - One or more LUNs for the SQL Server 2000 or 2005 database
  - One or more LUNs for the SnapInfo files

  For resource planning information, see Chapter 6, "Preparing to Migrate SQL Server Databases," on page 85. For detailed instructions for creating LUNs, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
- The databases to be migrated must not be in use. This includes both system databases and user databases.

  For information about how to check that users are not connected to a database, see your SQL Server documentation.
- The SQL Server databases to be migrated and the Data ONTAP LUNs to which they will be migrated must conform to SnapManager requirements, which include the following:
  - The LUNs used for the SQL Server database files cannot store any files other than those placed there by the SnapManager Configuration Wizard.
  - The LUN used for the SnapInfo director cannot store any files other than those placed there by the SnapManager Configuration Wizard.

  For more information, see "SQL Server database configurations to avoid" on page 88.

Overview of the database migration procedure

The following steps summarize the migration of SQL Server 2000 or 2005 database files:

1. The operator uses the Configuration Wizard to specify the databases to be migrated and the LUNs to which the databases are to be migrated.

   Note

   If the databases you intend to back up and restore using SnapManager are already on LUNs, and their configurations meet the requirements for operating with SnapManager, then you do not need to migrate them. Instead, use the Configuration Wizard only to set up the SnapInfo directory. No databases will be taken offline or copied.

   Note

   SnapManager for SQL 2.1 provides the capability to backup a Read-only database. Use the Configuration Wizard to migrate the Read-only database. However, you cannot select the Run UPDATE STATISTICS option for the Read-only database. During migration, SnapManager for SQL will skip this option only for Read-only database. After migration, you can restore and backup the Read-only database like any other normal database.

2. If you specified databases to be migrated to LUNs, the Configuration Wizard does the following:

   a. Detaches the specified databases.

   b. Copies the databases to the specified LUNs and sets up a SnapInfo directory.

      SnapManager detaches SQL Server *user databases* before migrating them to LUNs.

      SnapManager also stops the SQL Server while migrating SQL Server *system databases* to LUNs.

      Migrating SQL Server databases causes them to be taken offline during the copy operation.

      In a Windows cluster, if you migrate a database file to a Data ONTAP LUN that does not have dependency set on the SQL Server cluster resource, SnapManager places all resources directly or indirectly dependent on that LUN into an offline state while it adds the dependency on the cluster resource.

   c. Attaches the databases.

If a database copy or a database attach fails, SnapManager attaches the original database file to the SQL Server.

d. Deletes the old database files (if this was specified).

3. The Configuration Wizard sets up the SnapInfo directory structure per your specifications:

- Single SnapInfo Directory: Specify one SnapInfo directory for all SQL Server instances and their associated databases.
- Advanced SnapInfo Directories: For each SQL Server instance, specify a default SnapInfo directory for all the databases owned by that instance.

  If you have multiple databases that reside on one or two LUNs, specify a common SnapInfo directory for those databases.

  If you want to place the SnapInfo directory for an individual database on a LUN other than in the default location for that SQL Server instance, the Configuration Wizard supports the creation of that SnapInfo directory as well.

4. The Configuration Wizard reminds the operator to immediately back up the migrated databases.

Migrating system and resource databases

Use the Configuration Wizard to move the system databases to LUNs. While the Configuration Wizard is migrating SQL Server system databases to LUNs, the SQL Server is stopped by SnapManager.

Migrating SQL Server databases causes them to be taken offline during the move operation.

Run SnapManager Configuration wizard to move the master database. SnapManager for SQL also moves the resource database to the location where the master database is migrated.

Migrating databases to LUNs using the Configuration Wizard

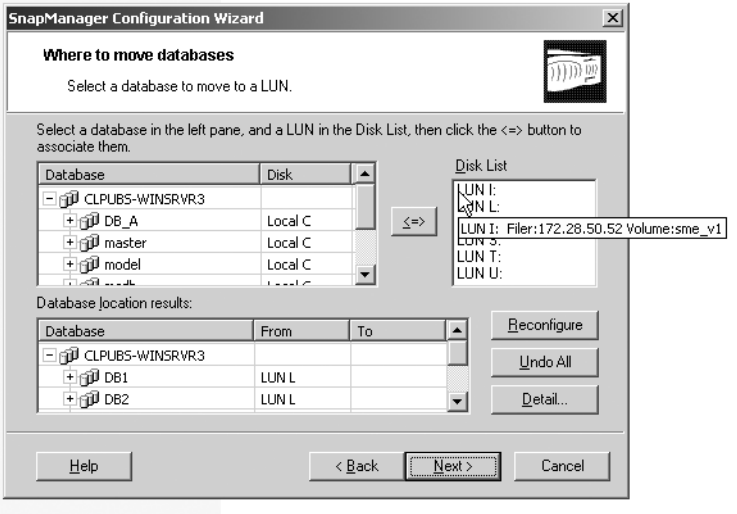
To migrate your SQL Server 2000 or 2005 databases from local disk to LUNs or from one LUN to another, complete the following steps.

Attention

After an SQL Server database has been migrated to LUNs using SnapManager, do not use the Enterprise Manager—or any other utility outside of SnapManager—to move any other data in the SQL Server system. Doing so might prevent SnapManager from functioning correctly.

Step	Action
1	If you have not already done so, start the SnapManager for SQL Server application. To do this, go to the Windows Start menu and select Program Files > SnapManager for SQL Server. **Result:** SnapManager for SQL Server main window opens.
Connect to SQL Server	
2	If you want to view or change the SQL Server managed by SnapManager, do this using the Connect to SQL Server dialog box. To open the dialog box, go to the SnapManager menu bar and select Action > Connect to a different SQL Server. For more information, see Appendix C, "Configuring SnapManager Application Settings," on page 363.
Start the Configuration Wizard	
3	Start the Configuration Wizard. ◆ From the SnapManager Wizards tab, click Configuration Wizard. ◆ From the SnapManager menu bar, select Wizard > Configuration Wizard. **Result:** The Configuration Wizard is launched and the Welcome screen appears.
Welcome	
4	At the Configuration Wizard Welcome screen, click Next. **Result:** The Database Verification Server and Account Authentication screen appears.

Step	Action
Database Verification Server and Account Authentication	
5	Use the Database Verification Server and Account Authentication screen to specify the SQL Server to be used to perform SQL Server database verification and account authentication. **Note** For optimal performance, use an SQL Server that is different from the verification server. ◆ If you want to specify the database verification server now, select an SQL Server in the Verification Server list and choose the security authentication method to be used to connect to that server. If you choose SQL Server authentication, you must also specify the login name and password. For a detailed description of these specifications, see "Using the Database Verification Server and Account Authentication screen" on page 408. ◆ If you want to specify the database verification server later, select the option labeled "Select a verification server later using the Options menu." Later, when you are ready to select a database verification server, you can do so by selecting Options > Database Verification Settings to open the Verification Settings dialog box. For a detailed description of the Verification Settings dialog box, see "Database integrity verification options" on page 372.
6	Click Next. **Result:** The Where to Move Databases screen appears.

Step	Action
Where to Move Databases	
7	Use the Where to Move Databases screen to specify the databases (or individual data files or transaction log files) you want to move and where you want them moved.  For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN. **Note** If the SnapManager host system is running SQL Server 2005 and the Disk List does not display any LUNs, be sure that the SQL Browser service was configured to start automatically. From this screen, you can do any of the following: ◆ Migrate databases from local disk to LUNs. See "Migrating SQL Server databases from local disk to LUNs" on page 411. ◆ Move databases from one LUN to another. See "Moving SQL Server databases to different LUNs" on page 413. ◆ View the path of a database that resides on a LUN. See "Viewing the full path for a database on a LUN" on page 416.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>8</td><td>Click Next.

Result: The Number of Databases that Can be Backed Up Concurrently screen appears and displays the current value.</td></tr>
<tr><td colspan="2">Number of Databases that Can be Backed Up Concurrently</td></tr>
<tr><td>9</td><td>Use the Number of Databases that Can be Backed Up Concurrently screen to change the maximum number of databases that can be backed up concurrently.

a. If you need to change the maximum number of databases that can be backed up concurrently for each SQL Server instance, change the value in the “Maximum databases...” box. The default setting is 255 databases.

b. If you also want to view and optionally change the maximum number of SQL Server worker threads for any SQL Server instance, select the Modify SQL Server “Maximum Worker Threads” Parameter option. This option is deselected by default.

For more detailed information, see “Using the Number of Databases that Can Be Backed Up Concurrently screen” on page 417.</td></tr>
<tr><td>10</td><td>Click Next.

Result: The next screen that appears depends on whether you chose to modify the SQL Server maximum number of worker threads allowed.
<table>
<tr><th>If...</th><th>Then...</th></tr>
<tr><td>You did not select the Modify SQL Server “Maximum Worker Threads” Parameter option</td><td>The SnapInfo Directory Type screen appears. Go to Step 13.</td></tr>
<tr><td>You selected the Modify SQL Server “Maximum Worker Threads” Parameter option</td><td>The Modify SQL Server Maximum Worker Threads Parameter screen appears. Go to Step 11.</td></tr>
</table></td></tr>
</table>

Step	Action
Modify SQL Server Maximum Worker Threads Parameter	
11	Use the Modify SQL Server Maximum Worker Threads Parameter screen to specify whether you want to modify the maximum number of worker threads for any SQL Server instance. You have the following options: ◆ Leave the current maximum number of worker threads unchanged for all SQL Server instances ◆ Apply all of the recommended new values displayed in the New Max Worker Threads column ◆ Edit one or more of the recommended new values before applying the changes For more detailed information, see "Using the Modify SQL Server Maximum Worker Threads Parameter screen" on page 419. **Note** If you choose to change the maximum number of worker threads for any SQL Server instances, those instances will be restarted when you click Finish in the Completing the Configuration Wizard screen.
12	After you have finished viewing and optionally changing this SQL Server parameter, click Next. **Result:** The SnapInfo Directory Type screen appears.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">SnapInfo Directory Type</td></tr>
<tr><td>13</td><td>Use the SnapInfo Directory Type screen to choose how you want the SnapInfo directory or directories configured.

Note
By default, the Single SnapInfo Directory option is selected. If SnapManager was previously configured with multiple SnapInfo directories, the SnapInfo Directory Type screen is displayed with the Advanced SnapInfo Directories option preselected.

<table>
<tr><th>If you want to…</th><th>Then do this…</th></tr>
<tr><td>Create one SnapInfo directory for all SQL Server instances and their associated databases (including multiple databases sharing one or two LUNs)</td><td>Go to Step 14.</td></tr>
<tr><td>Create a different SnapInfo directory for each SQL Server instance, or for multiple databases sharing one or two LUNs, or for each database in an SQL Server instance</td><td>Go to Step 18.</td></tr>
<tr><td>Move multiple SnapInfo directories to a single directory</td><td>Go to Step 18.</td></tr>
</table></td></tr>
<tr><td colspan="2">Specify a Single SnapInfo Directory</td></tr>
<tr><td>14</td><td>Select the Single SnapInfo Directory option, then click Next.

Result: The Specify a Single SnapInfo Directory screen appears.

Note
For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN.</td></tr>
</table>

Step	Action
15	Use the Specify a Single SnapInfo Directory screen to specify how you want to create a single SnapInfo directory for all your databases. By default, the SnapInfo directory name is `SMSQL_SnapInfo`. However, you can specify a different SnapInfo directory path by modifying the information in the Result SnapInfo Directory box. For more detailed information, see "Using the Specify a Single SnapInfo Directory screen" on page 423. **Result:** The Result SnapInfo Directory field displays the mount point for the newly specified SnapInfo directory.
16	Click Next. **Result:** The Database Migration Options screen appears.
17	Proceed to Step 25.
Where to Put the Default SnapInfo Directory for Each SQL Server Instance	
18	Select the Advanced SnapInfo Directories option, then click Next. **Result:** The Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen appears. **Note** For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN.

Step	Action
19	Use the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen to create a default SnapInfo directory for all the databases owned by a specific SQL Server instance. In subsequent screens, you will have the opportunity to override this setting for specific cases: ◆ Multiple databases that share one or two LUNs ◆ A single database **Note** You can also use this screen to change from multiple SnapInfo directories to a single SnapInfo directory. By default, the SnapInfo directory name is `SMSQL_SnapInfo`. However, you can specify a different SnapInfo directory path by using the Default SnapInfo Property dialog box. For more detailed information, see "Using the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen" on page 425.
20	Click Next. **Result:** The Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen appears. **Note** For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN.

Step	Action
Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases	
21	If you have multiple databases that reside on one or two LUNs, use the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen to specify a common SnapInfo directory location for those databases. By default, the SnapInfo directory name is `SMSQL_SnapInfo`. However, you can specify a different SnapInfo directory path by using the Default SnapInfo Property dialog box. For more detailed information about using the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen, see "Using the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen" on page 428.
22	Click Next. **Result:** The Where to Put the SnapInfo Directory for Each Database screen appears. **Note** For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN.

Step	Action
Where to Put the SnapInfo Directory for Each Database	
23	If you want to place the SnapInfo directory for an individual database on a LUN other than in the default location for that SQL Server instance, use the Where to Put the Default SnapInfo Directory for Each Database screen to select the database and specify the SnapInfo directory location. By default, the SnapInfo directory name is `SMSQL_SnapInfo`. However, you can specify a different SnapInfo directory path by using the Default SnapInfo Property dialog box. For more detailed information about using the Where to Put the Default SnapInfo Directory for Each Database screen, see "Using the Where to Put the SnapInfo Directory for Each Database screen" on page 431.
24	Click Next. **Result:** The Database Migration Options screen appears. **Note** The next time you run the Configuration Wizard, the Advanced SnapInfo Directories option will be preselected (in the SnapInfo Directory Type screen) if the previous configuration operation was performed with two or more SnapInfo directories.
Database Migration Options	
25	Use the Database Migration Options screen to select options for the database migration: ◆ When you want to run the DBCC command to verify your migrated databases ◆ Whether you want to delete the original database files after the migration finishes successfully ◆ Whether you want to run UPDATE STATISTICS on tables before detaching the databases For a more detailed description, see "Using the Database Migration Options screen" on page 434.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>26</td><td>Click Next.

Result: The next screen that appears depends on whether SnapManager is installed in an MSCS environment or on a stand-alone Windows host.<table><tr><th>If...</th><th>Then...</th></tr><tr><td>You are configuring SnapManager in a MSCS environment -or- the Microsoft iSCSI service is not running on the Windows host system</td><td>The Configure Automatic Event Notification screen appears. Go to Step 29.</td></tr><tr><td>You are configuring SnapManager on a stand-alone Windows host system</td><td>The Add Microsoft iSCSI Service Dependency screen appears. Proceed to Step 27.</td></tr></table></td></tr>
<tr><td colspan="2">Add Microsoft iSCSI Service Dependency</td></tr>
<tr><td>27</td><td>Use the Add Microsoft iSCSI Service Dependency screen to optionally configure the Microsoft iSCSI service as a dependency for the Microsoft SQL Server services on a particular SQL Server instance.

For each SQL Server instance, if you are migrating databases to LUNs accessed using the iSCSI protocol, then select the option to ensure that the Microsoft iSCSI service will be added as a dependency for the Microsoft SQL Server services in that instance. This will prevent Microsoft SQL Server from starting before the Microsoft iSCSI service is ready to accept connections.

Note
You can click Select All to select every option in the list. You can click Unselect All to clear every option.</td></tr>
<tr><td>28</td><td>Click Next.

Result: The Configure Automatic Event Notification screen appears.</td></tr>
</table>

Step	Action
Configure Automatic Event Notification	
29	Use the Configure Automatic Event Notification screen to configure the automatic event notification options for SnapManager: ◆ Configure automatic SMTP e-mail notification for event notification. ◆ Enable Syslog (storage system event log) support. ◆ Enable AutoSupport for SnapManager. ◆ Limit event logging to failure events **Note** NetApp strongly recommends that you enable the automatic event notification options. After you complete the Configuration Wizard, you can also view or change these settings by selecting Options > Notification Settings to access the Auto Notification Settings dialog box. For more detailed information, go to "Event notification options" on page 401. **Result:** The Completing the Configuration Wizard screen appears.
Completing the Configuration Wizard	
30	In the Completing the Configuration Wizard screen, click Finish to close the Configuration Wizard so that you can begin the actual database configuration process. **Note** If you chose to change the maximum number of worker threads for any SQL Server instances (specified in the Modify SQL Server Maximum Worker Threads Parameter screen), those instances will be restarted when you click Finish. **Result:** The Configuration Wizard closes, and the Configurator Status dialog box appears.

Step	Action
Configurator Status	
31	To migrate your databases to the LUNs you specified and to set up the SnapInfo directory as specified, click Start Now. **Result:** The Configuration Wizard moves the databases and logs to their specified LUNs and displays a task check-off list. ◆ Migrating SQL Server databases causes them to be taken offline during the copy operation. ◆ In a Windows cluster, if you migrate a database file to a Data ONTAP LUN that does not have dependency set on the SQL Server cluster resource, SnapManager adds the LUN dependency on the cluster resource. **Note** It is a Windows cluster requirement that a resource must be taken offline prior to adding a dependency to it. In order to add this LUN dependency on the cluster resource, SnapManager places all resources directly or indirectly dependent on that LUN into an offline state, adds the dependency on the cluster resource, then brings the resources back online. ◆ If a database copy or a database attach fails, SnapManager attaches the original database file to the SQL Server. When the migration finishes successfully, SnapManager displays a message box. This message also reminds you to make a backup immediately.
32	Click OK to close the message box.
33	Back up your databases immediately. Any time you use the Configuration Wizard to move data from one drive to another, backups performed prior to the move become invalid. After you migrate your databases to a new LUN, perform an immediate backup.

Moving multiple SnapInfo directories to a single SnapInfo directory

If you previously configured multiple SnapInfo directories, you can rerun the Configuration Wizard to move them to a single SnapInfo directory.

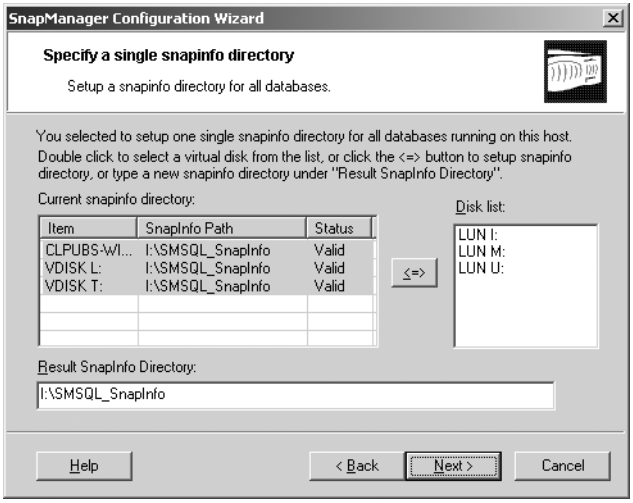
The Configuration Wizard enables you to create multiple SnapInfo directories in the following ways:

- A default SnapInfo directory per SQL Server instance
- A separate SnapInfo directory for multiple databases on one or two LUNs
- A different (nondefault) SnapInfo directory for a database in an instance

If you currently have multiple SnapInfo directories, you can choose to combine them into a single directory.

To change your configuration from multiple SnapInfo directories to a single SnapInfo directory for all SQL Server instances and associated databases, complete the following steps.

Step	Action
1	Start the Configuration Wizard, then step through the following screens without specifying any configuration changes: ◆ Database Verification Server and Account Authentication ◆ Where to Move Databases ◆ Number of Databases that Can be Backed Up Concurrently

Step	Action
SnapInfo Directory Type	
2	In the SnapInfo Directory Type screen, select the Single SnapInfo Directory option, and then click Next.  **Result:** The Specify a Single SnapInfo Directory screen appears. Note that, in the Current SnapInfo Directory list, all of the current SnapInfo directories are selected by default.
Specify a Single SnapInfo Directory	
3	In the Disk List, select the LUN to which you want to move all the SnapInfo directories. **Note** For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN.
4	Click the move (<=>) button. **Result:** The Result SnapInfo Directory box displays the path for the SnapInfo directory. Note that the default directory name is `SMSQL_SnapInfo`.

Step	Action
5	If you want to specify a different location or name, modify the information in the Result SnapInfo Directory box. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see "SQL Server database configuration restrictions enforced" on page 87.
Complete the Configuration Wizard	
6	Step through the remaining screens of the Configuration Wizard without specifying any further configuration changes: ◆ Database Migration Options ◆ Add Microsoft iSCSI Service Dependency ◆ Configure Automatic Event Notification
7	In the Completing the Configuration Wizard screen, verify the changes you specified, and then click Finish.
8	In the Configurator Status dialog box, click Start Now to close the dialog box, and then begin the Configuration Wizard tasks you specified.
9	When a message box appears and notifies you that the configuration changes were completed successfully, click OK to close the message box.
10	Click Close to close the Configurator Status dialog box.

Migrating SQL Server databases back to local disks

If, for any reason, you choose not to use SnapManager as your data management tool, you can migrate your databases back to local disks.

To migrate your databases back to local disks, complete the following steps.

Step	Action
1	From the SnapManager menu bar, select Options > Configuration Wizard Options > Migrate Database(s) to Local Disk. **Result:** A dialog box appears and informs you that the migrate back to local disk feature has been enabled. **Note** By default, the Where to Move databases screen of the Configuration Wizard does not list any local drives unless you explicitly enable the migration to local disk feature.
2	Click OK.
3	Launch the SnapManager Configuration Wizard.
4	Click Next at the Welcome screen.
5	In the Where to Move Databases screen, select the database that you want to move back to local disk from the Database Location Results list. ◆ To select multiple databases or files, select the first entry, and then click and hold the Shift button on your keyboard while you make additional selections. ◆ To select a range of databases or files, select the first entry in the range, and then click and hold the Ctrl button on your keyboard while you select the last entry in the range.
6	Click Reconfigure.

Step	Action
7	In the database list, select the database or databases you just reconfigured. **Note** In the "Select a Database..." list, the Disk column for this entry lists `Reconfig` instead of the LUN drive letter.
8	In the LUN list, select a local drive, and then click the <=> button.
9	Click Next.
10	In the Where to Put the Default SnapInfo File screen, click Next. **Note** Both SnapInfo directories must remain on the LUNs on which you placed them during the original migration. They cannot be moved to local disk.
11	In the Database Migration Options screen, click Next.
12	In the Completing the Configuration Wizard screen, click Finish.
13	Click Start Now to migrate your databases back to local disk.

Understanding SnapManager Backup Sets 8

This section provides background information about SnapManager Backup and SnapManager backup sets. The following topics are covered:

- "How SnapManager Backup works" on page 130
- "Types of backup operations performed using SnapManager" on page 132
- "How SnapManager checks database integrity in backup sets" on page 135
- "How SnapManager backup data is organized" on page 139
- "Ways to manage the number of backup sets kept online" on page 144
- "When to run a SnapManager backup" on page 147

Note

If you are running SQL Server 2000, SnapManager does not allow you to restore a backup set created using SQL Server 2005.

How SnapManager Backup works

SnapManager Backup uses Snapshot copy functionality to create online, read-only copies of the SQL Server data you specify.

Note

Databases that cannot be backed up by SnapManager are listed in the upper left panel of the Backup and Verification tab or the Backup Wizard with the label `Invalid` next to the LUN drive letter.

See the following topics for more information:

- "What SnapManager Backup does" on page 130
- "SnapManager Backup requirements and limitations" on page 131

What SnapManager Backup does

SnapManager for SQL Server performs the following tasks when creating a backup:

1. Checks the SnapManager license
2. Renames the most recent SnapInfo directory (if necessary)
3. Renames the most recent Snapshot copy (if necessary)
4. Creates a new directory in the SnapInfo directory for this backup

   Note

   During the backup process, SnapManager collects backup metadata that is automatically archived to the SnapInfo directory.

5. Creates a Snapshot copy of the LUNs containing the database files (if specified)
6. Backs up transaction logs (if specified)
7. Creates a Snapshot copy of the LUN that contains the SnapInfo directory
8. Verifies the databases in the Snapshot copy of the LUN (if specified)
9. Deletes the oldest backup sets (if specified)
10. Deletes the oldest Snapshot copy of the LUN that contains the SnapInfo directory

SnapManager Backup requirements and limitations

Be aware of the requirements and limitations of SnapManager Backup:

- To run SnapManager Backup, the account that SnapManager is using must have a system administrator server role on the SQL Server.
- If you rename the SQL database and then need to restore the database from a backup set that was created before the database was renamed, you must restore to a different and nonexistent database name.
- If you change the database configuration, any backups taken prior to the configuration change are invalid. After you run the Configuration Wizard, immediately take a backup to reflect the change in your configuration.

Attention

In an SQL Server 2000 or 2005 environment, you should perform backups using only the SnapManager application. Making Snapshot copies of the storage system console for backup using SnapManager is not supported and results in an inconsistent Snapshot copy image of online databases. However, you can use SnapDrive to make Snapshot copies of SQL Server databases, although you cannot restore these Snapshot copies using SnapManager.

Types of backup operations performed using SnapManager

SnapManager supports two types of backup operations:

- "Full database backup" on page 132
- "Transaction log backup" on page 134

Note

A transaction log backup can be included in a full database backup, or it can be created as a log-only backup set.

Full database backup

A full database backup contains a full copy of the databases you select. The method that SnapManager uses to create the backup depends on the databases you select. One method involves copying the databases individually, while the other method consists of creating Snapshot copies of the databases. The method that SnapManager uses to create a particular backup set has implications for how SnapManager restores databases from that backup set. For more information, see "How SnapManager Restore works" on page 219.

Stream-based backup method: With this method, SnapManager creates the full database backup by *streaming out the contents of the databases individually.* SnapManager uses the stream-based method to back up the following:

- All *system databases*
- Any *user databases* that reside on the same LUN as a system database

All other database backups use the *online Snapshot* backup method.

For SQL Server 2005, if there is no system database on the SQL Server *root LUN*, SnapManager cannot back up the user databases on that LUN and lists those user databases with the label `Invalid` next to the LUN drive letter.

Note

A user database cannot reside on a LUN which is also used by SQL root path, SQL data root path, or SQL error log path unless there is a system database on that LUN. This restriction is enforced by the Configuration Wizard.

Full database stream-based backup files are `.fbk` files named using the convention *date_time_databasename*: for example, `050802_0330_xxx.fbk`. This file is equivalent to the `.bak` file directly created by SQL Server 2000 or 2005.

Online Snapshot backup method: With this method, SnapManager creates the backup by *creating Snapshot copies of the databases*. SnapManager uses the online Snapshot method to back up *user databases* that do not reside on the same LUN as any system databases. All other database backups use the *stream-based* backup method.

When you select a database for a full database backup, SnapManager automatically selects all other databases that reside on the same storage system volume. You can deselect databases that reside on a different LUN from the database(s) you selected, even if the LUN is on the same storage volume. If the other LUN stores only a single database, you can deselect or reselect that database individually. If the other LUN houses multiple databases, you must deselect or reselect those databases as a group.

For a description of the naming convention used by full database online Snapshot backup sets, see "SnapManager backup set names" on page 141.

More about volume-wide backups: In a volume-wide backup, all of the databases that reside on a single volume are backed up *concurrently.*

- If the databases reside on the same volume, only one backup Snapshot copy is created.
- If the databases reside on multiple volumes, more than one Snapshot copy is created; the databases are grouped by volume and backed up *in successive order*, one Snapshot copy per volume.

Note

When a Snapshot copy is made of a LUN for a SnapManager backup, the entire storage system volume is captured in that Snapshot copy. However, that backup is valid only for the SQL host server for which the backup was created. If data from other SQL host servers resides on the same volume, that data is not restorable from the Snapshot copy.

About Enterprise Manager and Management Studio: Although SnapManager *Snapshot copy* full database backup files are viewable from the Enterprise Manager or Management Studio of your SQL Server, you cannot perform any operations on them using the SQL Server backup utility.

Transaction log backup

A transaction log backup is a record of the database changes that have occurred since the last transaction log backup. SnapManager supports transaction log backups in order to provide a more granular level of database backup and to recover the transactions committed since the most recent full backup.

File name and location: SnapManager creates a backup of a transaction log by copying transaction log data to a file in the SnapInfo directory. Transaction log backup files are named using the following convention:

date_time_databasename.trb

This file is equivalent to the .trn file directly created by SQL Server. The structure of the SnapInfo directory is described in "How SnapManager backup data is organized" on page 139.

Ways to start or schedule a transaction log backup: You can back up a transaction log along with the database or alone.

- SnapManager full database backups include the option to also back up the associated transaction logs after the database Snapshot copy backups finish. This is described in "Creating a full database backup using SnapManager" on page 152.
- SnapManager also provides the option to back up transaction logs only, independent of the associated databases. This is described in "Creating a transaction log backup using SnapManager" on page 172.

About log shipping and other backup solutions: NetApp recommends that you use SnapManager alone to back up your SQL Server database transaction log files. If you decide to use a different backup solution, use it alone as well; do not attempt to restore from backup files that were created using different backup solutions.

If *log shipping* is implemented for a particular database, heed the following recommendations:

- When using SnapManager Backup, do not back up the transaction logs for that database.
- When using SnapManager Restore to restore that database, (1) disable the option to create a transaction log backup before the restore and (2) do not restore the transaction logs.

About Enterprise Manager and Management Studio: SQL Server Enterprise Manager and Management Studio both detect transaction log backups taken by SnapManager for Microsoft SQL Server and restores the database to a further point in time by applying transaction log backups in sequence. However, neither Enterprise Manager nor Management Studio can restore full database backups of Snapshot copies made by SnapManager for Microsoft SQL Server.

How SnapManager checks database integrity in backup sets

SnapManager uses Database Consistency Checker (DBCC) to verify SQL Server databases. DBCC is a Microsoft SQL Server utility that verifies the page-level integrity of databases. See the following topics for more information:

- "Ways that SnapManager uses SQL Server DBCC" on page 135
- "Drive letter requirements for running SQL Server DBCC against the databases in a backup set" on page 136
- "Ways to separate database verification from database backup" on page 137
- "Options for when to verify the databases in a backup set" on page 137
- "Options for where to run SQL Server DBCC" on page 138
- "Verifying backup sets in a mixed SQL Server environment" on page 138

Ways that SnapManager uses SQL Server DBCC

SnapManager uses the DBCC CHECKDB command to verify the integrity of live databases as well as databases in SnapManager backup sets.

Verifying the integrity of live databases: Live databases can be verified as a part of database migration and also as a part of a full database backup.

- Using the Configuration Wizard, you can verify live databases before and after database migration. For more information, see "Using the Database Migration Options screen" on page 434.
- Using SnapManager Backup, you can verify live databases before and after a full database backup. For more information, see "Configuring the profile of a full database backup" on page 378.

Verifying the integrity of databases in backup sets: Databases in backup sets can be verified upon creation, separately, or before a restore.

- Using SnapManager Backup, you can verify the databases in full database backup sets as they are created or you can verify the databases in the most recent unverified backup sets.
- Using SnapManager Restore, if you select a backup set on which a consistency check has not been run successfully, SnapManager prompts (but does not require) you to first verify the databases in that backup set.

Attention

The SnapManager Restore tab lists the backups that have been taken and the backup verification status of each. For more information, see "Importance of verifying databases to be restored" on page 221.

Drive letter requirements for running SQL Server DBCC against the databases in a backup set

When you verify the databases in a backup set (as opposed to live databases), Microsoft DBCC requires that all of the databases files are mounted at the same time. At a more granular level this means that SnapManager, using SnapDrive commands, mounts all the LUNs that contain the backup sets selected for database verification.

Each LUN that will be mounted requires one available drive letter: In order to run the DBCC CHECKDB command, the verification server (whether local or remote) must have a sufficient number of drive letters available to mount all the LUNs that store the database backup sets you are verifying.

- When you run database verification against backup sets that are stored on *a single LUN*, the SQL Server computer that is used as the verification server must have at least *one drive letter available* so that the LUN can be mounted during database verification.
- When you run database verification against backup sets that contain *multiple database files stored on separate LUNs*, SnapManager mounts all of these LUNs at the same time. Consequently, the SQL Server computer that is used as the verification server must have *enough drive letters available* so that SnapManager can mount each of the LUNs simultaneously.

  For example, suppose you want to run database integrity verification against backup sets containing five file groups using three transaction logs stored on on eight separate LUNs. In this case, the verification server would need to have a minimum of eight drive letters available.

Lack of available drive letters causes DBCC CHECKDB to fail: If the verification server runs out of available drive letters while attempting to run DBCC CHECKDB for a SnapManager operation, the SnapManager operation will fail with the following message in the report log:

```
[SnapDrive Error]: There are no remaining drive letters available
on the system.  Please delete or disconnect a drive and retry.
```

The SnapManager operations that enable you to verify the databases in backup sets are as follows:

- Full database backup with verification of the databases in the backup set. For detailed information, see "Creating a full database backup using SnapManager" on page 152.
- Verification of the databases in the most recent unverified backup sets. For detailed information, see "Performing database verification using SnapManager" on page 187.
- Verification of the databases in an unverified backup set selected for a restore operation. For detailed information, see "Performing a restore operation" on page 225.

Ways to separate database verification from database backup

Running database verification on a production SQL Server is CPU-intensive for the host running the verification and also involves a substantial amount of activity on the storage system. For this reason, verification can degrade SQL Server response, particularly during work hours.

By default, a SnapManager full database backup operation runs DBCC immediately after the backup is finished. However, SnapManager provides the two options that enable you to separate the process of verification from the backup itself: deferred database verification and remote database verification.

Deferred database verification: Instead of allowing a full database backup to automatically verify the databases when the operation is finished, you can disable that feature. You can then run a separate database verification operation any time after the full database backup operation is finished.

Remote database verification: To prevent database verification from affecting the performance of your production SQL Server computer, you can run verification on another SQL Server computer.

Options for when to verify the databases in a backup set

You can verify the databases in your SnapManager backup sets at various times.

Automatically verify full database backup sets upon creation: By default, SnapManager verifies the databases in a backup set at the time the backup is created. This is simple and ensures that each database in the backup set is verified. However, this method significantly increases the time required to complete the backup. For a detailed description of how to start or schedule a full database backup with automatic database verification, see "Creating a full database backup using SnapManager" on page 152.

Explicitly start or schedule database verification only: With this method, a single operation can be initiated to verify the databases contained in one or more backup sets that have already been created. You can start the verification immediately, or you can schedule the verification to occur later, when it does not affect performance or delay later backups. For a detailed description of how to start or schedule database verification, see "Performing database verification using SnapManager" on page 187.

Defer verification until you restore from the backup set: If you attempt to restore from a backup set on which a database consistency check has not been run successfully, SnapManager prompts (but does not require) you to first verify the databases in that backup set. See "Importance of verifying databases to be restored" on page 221.

Options for where to run SQL Server DBCC

Regardless of when you verify the databases in a backup set, the verification can be done on the production SQL Server (the Windows host system running the SQL Server instance used to create the databases) or on a remote verification system (another SQL Server).

From the production SQL server: In the simplest SnapManager for SQL Server configuration, verification is run from the production SQL Server. However, because the Microsoft DBCC command used for the verification is CPU-intensive, performing the verification on the production SQL Server host system during peak usage could affect SQL Server performance.

From a remote verification server: Performing the verification on a remote system minimizes the impact of verification on SQL Server system resources and backup schedule. The requirements for a remote verification server are described in "Requirements for a remote verification server" on page 29. The procedure specifying a different SQL Server as the remote verification server is described in "Selecting the database verification server" on page 372.

Note

You cannot run database verification from a virtual SQL Server. For more information, see "Requirements for a remote verification server" on page 29.

Verifying backup sets in a mixed SQL Server environment

Be aware of how SQL Server DBCC handles backward-compatibility between SQL Server versions of the *verification server* and SQL Server versions of the *databases in the backup set*.

- If you use an *SQL Server 2005 verification server* to verify an SQL Server 2000 database, SQL Server upgrades a copy of the database to make it compatible with the SQL Server 2005 DBCC CHECKDB command.

  Note

  This combination is not optimal, because additional processing overhead is incurred *each time* an SQL Server 2000 database backup set is verified using an SQL Server 2005 verification server.

- If you use an *SQL Server 2000 verification server* to verify an SQL Server 2005 database, SQL Server DBCC will fail.
- If you select an *SQL Server 2005 database* for verification, SnapManager prevents you from selecting an SQL Server 2000 verification server for that operation.

How SnapManager backup data is organized

SnapManager backup data is stored in backup sets. See the following topics for more information:

- "SnapManager backup sets" on page 139
- "SnapInfo directory" on page 139
- "SnapInfo subdirectory names" on page 140
- "SnapManager backup set names" on page 141
- "SnapManager Snapshot naming conventions" on page 142
- "SQL Server database Snapshot names" on page 143
- "SnapInfo directory Snapshot names" on page 143

SnapManager backup sets

A SnapManager backup set consists of all the data you would need to be able to perform a restore, regardless of whether this data exists on the same LUNs or volumes. A backup set contains the following items:

- Volume-wide Snapshot copies of the LUNs that store the SQL Server databases (SQL Server data files and SQL Server transaction log files)
- The corresponding subdirectory under the SnapInfo directory

SnapInfo directory

The SnapInfo directory stores information about the backup sets, copies of transaction log files, and other data critical to the backup set. The location of this directory is specified when you run the Configuration Wizard. By default, the directory name is `SMSQL_SnapInfo`. However, you can specify a different directory name.

Every time a SnapManager backup set is created, SnapManager creates a new backup set subdirectory under the SnapInfo directory. SnapManager populates this subdirectory with the transaction logs backed up as part of that backup set, as well as the recovery information for that specific Snapshot copy. A complete backup set consists of this SnapInfo subdirectory and the corresponding Snapshot copies of the LUNs that store the SQL Server databases and transaction logs.

Note

The SnapInfo directory cannot reside on the same LUN that stores the SQL Server database files. See "SQL Server database configurations to avoid" on page 88.

SnapInfo subdirectory names

SnapManager backup set names identify the configuration of the backed-up databases.

Configuration	Format of the SnapInfo subdirectory name
Databases belonging to the SQL Server default instance	The SnapInfo directory name is `SQL__` followed by the SQL Server computer host name: `SQL__`*SqlServerName* For example, the subdirectory for databases belonging to the default instance of the SQL Server on the Windows host system `CLPUBS-WINSRVR3` would be named as follows: `SQL__CLPUBS-WINSRVR3`
Databases belonging to an SQL Server named instance	The SnapInfo directory name is `SQL__` followed by the name of the SQL Server instance: `SQL__`*InstanceName* For example, the subdirectory for databases belonging to the SQL Server instance `INST2` on the on the Windows host system `ENGR-WINSRVR7` would be named as follows: `SQL__INST2`
Multiple databases sharing one LUN	The SnapInfo directory name incorporates the drive letter assigned to the LUN: `VDisk__`*DriveLetter* For example, the subdirectory for all the databases sharing one LUN on drive `G:\` would be named as follows: `VDisk__G`
Multiple databases sharing two LUNs	The SnapInfo directory name incorporates both the drive letters assigned to the two LUNs: `VDisk__`*FirstDriveLetter&SecondDriveLetter* For example, the subdirectory for all the databases sharing the two LUNs on drive `F:\` and drive `K:\` would be named as follows: `VDisk__F&K`

SnapManager backup set names

SnapManager backup set names identify the configuration of the backed-up databases. These names are displayed in the SnapManager Restore tab and in the SnapManager Restore Wizard.

Configuration	Format of the backup set name
Databases belonging to the SQL Server default instance	The backup set name is the same as the SQL Server computer host name: *SqlServerName* For example, a backup set for databases belonging to the default instance of the SQL Server on the Windows host system `CLPUBS-WINSRVR3` would be named as follows: `CLPUBS-WINSRVR3`
Databases belonging to an SQL Server named instance	The backup set name is the name of the SQL Server instance: *InstanceName* For example, a backup set for databases belonging to the SQL Server instance `INST2` on the on the Windows host system `ENGR-WINSRVR7` would be named as follows: `INST2`
Multiple databases sharing one LUN	The backup set name incorporates the drive letter assigned to the LUN: `VDisk__`*DriveLetter* For example, a backup set for all the databases sharing one LUN on drive `G:\` would be named as follows: `VDisk__G`
Multiple databases sharing two LUNs	The backup set name incorporates both the drive letters assigned to the two LUNs: `VDisk__`*FirstDriveLetter&SecondDriveLetter* For example, a backup set for all the databases sharing the two LUNs on drive `F:\` and drive `K:\` would be named as follows: `VDisk__F&K`

SnapManager Snapshot naming conventions

The Snapshot copies created by SnapManager backup operations are automatically named by Data ONTAP. The name of each Snapshot copy created during a SnapManager backup operation includes information that identifies the Snapshot copy contents.

SQL Server name: Database Snapshot copy names and SnapInfo directory Snapshot copy names include the name of the SQL Server for which the backup was made (indicated in this document by the variable *SqlServerName*).

Backup management group: Database Snapshot copy names and SnapInfo directory Snapshot copy names include the backup management group to which you assigned the full database backup. SnapManager provides backup management groups for designating various levels of backup retention: Standard, Daily, and Weekly:

- If you assign a full database backup to the *Standard* backup management group, the Snapshot copy names for the databases and SnapInfo directory do not include a backup management group name.
- If you assign a full database backup to the *Daily or Weekly* management groups, the Snapshot copy names for the databases and SnapInfo directory include the name of the backup management group (indicated in this document by the variable *BackupMgmtGrp*).

For more information about using backup management groups, see "Using backup management groups in backup and verification" on page 198.

Most recent backup: SnapManager 1.0 for Microsoft SQL Server always appended the string `recent` to the name of the most recently created Snapshot copy. This was to allow external scripts—for example, archive scripts— to be able to identify and operate on the most recent Snapshot copy. With the addition of the Run Command After Operation feature in SnapManager 1.5 for Microsoft SQL Server, appending `recent` is no longer necessary because the scripts can be integrated into the backup process.

Like SnapManager 1.5 and 2.0 for Microsoft SQL Server, SnapManager 2.1 for Microsoft SQL Server offers two conventions for naming backup Snapshot copies:

- **Unique backup naming.** The most recent Snapshot copy name contains the Snapshot copy creation date and time (indicated by the variable *date_time*) instead of the string `recent`. The most recent backup is identified by the Snapshot copy name with the most recent date and time.
- **Generic backup naming.** The most recent Snapshot copy name contains the string `recent` instead of a date and timestamp. The most recent backup is identified by the Snapshot copy name that includes the string `recent`. This is

the Snapshot copy naming convention used by SnapManager 1.0 for Microsoft SQL Server.

The backup naming convention is selected in the Full Database Backup tab of the Backup Setting dialog box. For information about using this dialog box, see "Configuring the profile of a full database backup" on page 378.

SQL Server database Snapshot names

For SQL Server database backups, Snapshot copy names begin with the string `sqlsnap__`.

Backup management group	Format of the SQL Server database Snapshot copy name
Standard	Depending on the naming convention selected: ◆ `sqlsnap__`*SqlServerName_date_time* ◆ `sqlsnap__`*SqlServerName*`__recent`
Daily or Weekly	Depending on the naming convention selected: ◆ `sqlsnap__`*SqlServerName_date_time__BackupMgmtGrp* ◆ `sqlsnap__`*SqlServerName*`__recent`

SnapInfo directory Snapshot names

For SnapInfo directory backups, Snapshot copy names begin with the string `sqlinfo__`.

Backup management group	Format of the SnapInfo directory Snapshot copy name
Standard	Depending on the naming convention selected: ◆ `sqlinfo__`*SqlServerName_date_time* ◆ `sqlinfo__`*SqlServerName*`__recent`
Daily or Weekly	Depending on the naming convention selected: ◆ `sqlinfo__`*SqlServerName_date_time__BackupMgmtGrp* ◆ `sqlinfo__`*SqlServerName*`__recent`

Ways to manage the number of backup sets kept online

When planning your SnapManager backup schedules, you also need to manage the number of Snapshot-based backup sets that are stored online. See the following topics for more information:

- "Maximum number of Snapshot copies per volume" on page 144
- "Automatic deletion of the oldest backups in a management group" on page 144
- "Explicit deletion of backup Snapshot copies" on page 146

Maximum number of Snapshot copies per volume

The Data ONTAP software used with SnapManager 2.1 for Microsoft SQL Server supports a maximum of 255 Snapshot copies per storage system volume. Snapshot copies not created by SnapManager also count toward this total.

Because each SnapManager backup operation creates Snapshot copies, a SnapManager backup operation will fail if you have already saved 255 database Snapshot copies to disk and have not removed any of the Snapshot copies.

Note

It is possible for the total number of Snapshot copies on a volume to exceed the number of SnapManager backups being retained. For example, if a single volume contains both the SnapInfo directory and the SQL Server databases, each SnapManager backup generates two Snapshot copies on that volume.

To avoid reaching the limit of 255 Snapshot copies per volume, delete your old SnapManager backups when they are no longer needed. SnapManager provides the following ways to manage and delete backups:

- Automatic deletion
- Explicit deletion

These two methods are described in-depth in the following subsections.

Automatic deletion of the oldest backups in a management group

When you start or schedule a full database backup, you can also specify the number of backups of that database to be retained for that backup management group. After the backup is finished, SnapManager will automatically delete the oldest backups for that database in the specified backup management group, retaining only the number of backups you want to preserve.

This is the recommended method for managing the number of Snapshot copies you store on your system.

The procedural details are included in "Creating a full database backup using SnapManager" on page 152 and "Creating a transaction log backup using SnapManager" on page 172. In summary, you can enable automatic deletion of backups by specifying the "Delete Oldest Backups In Excess Of" option in either the Backup and Verification tab or one of the SnapManager Backup wizard screens. If you select this option, then after the backup is finished, SnapManager automatically deletes the oldest backups of the database from the same backup management group so that only the number of backups you specify are preserved.

For more information about backup management groups, see "SnapManager Snapshot naming conventions" on page 142 and "Using backup management groups in backup and verification" on page 198.

Cases in which additional backups are preserved: SnapManager does not count backups that failed verification when counting the number of stored backups. Therefore, more backups might be preserved than you specify in the "Delete Oldest Backups In Excess Of" box.

For example, suppose you are backing up databases A and B, which contain the following backup sets.

SnapManager backup set	Description
Database A	
`sqlsnap__orbit3_11-23-2004_16.21.07`	Old backup—good
`sqlsnap__orbit3__recent`	Recent backup—good
Database B	
`sqlsnap__orbit3_11-23-2004_16.21.07`	Old backup—good
`sqlsnap__orbit3__recent`	Recent backup—inconsistent

Also suppose you have set the "Delete Oldest Backups in Excess Of" box to `1` to preserve only one of each backup set, the most recent one.

In order to preserve one good backup for Database B, SnapManager will not delete the Snapshot copy `sqlsnap__orbit3_11-23-2004_16.21.07`. Therefore, two backups for Database B remain instead of one.

Option to retain up-to-the-minute restore ability: If you delete backups that are not the oldest backups in your backup list, and the corresponding transaction logs are also deleted, this makes the older remaining backups no longer available for an up-to-the-minute restore. The reason is that the transaction logs are no longer contiguous from the time when the older backup was taken to the present time.

This can happen when you are deleting backups of a particular backup management group.

SnapManager for Microsoft SQL Server enables you to preserve the logs in this case, thereby retaining the ability to use the older backups in an up-to-the-minute restore.

Note

If you know you will not need to perform an up-to-the-minute restore from the older backups, then you should allow the logs to be deleted in order to free up more space on the storage system holding the backups.

Explicit deletion of backup Snapshot copies

In addition to *automatically* deleting the oldest backup Snapshot copies (an option that you can select when you start or schedule a backup operation), you can *explicitly* delete individual or multiple backup Snapshot copies.

Explicit deletion of an individual backup: With this method, you delete individual selected Snapshot copies for either full database backups or transaction logs. The procedures are described in "Deleting a specific full database backup set" on page 207 and "Deleting a specific transaction log backup" on page 209.

Explicit deletion of multiple backups: With this method, you select a database to be deleted, the types of backup set components to be deleted (full database backups, transaction logs only, or SnapInfo directory backups), and the type of backup management group to be deleted. The procedure is described in "Deleting multiple backups" on page 203.

Note

You can also explicitly delete the LUN Snapshot copies that were created during a restore operation. For a description of *restore Snapshot copies*, see "How SnapManager Restore works" on page 219. For a description of how to view and delete these Snapshot copies, see "Deleting restored Snapshot copies" on page 247.

When to run a SnapManager backup

You need to balance frequency of backups against any overhead incurred by the database verification process. In addition, you must ensure that no SnapManager operations overlap with each other. See the following topics for more information:

- "Backing up databases following data migration" on page 147
- "Best time to run a SnapManager backup" on page 147
- "Frequency of backups" on page 147
- "Recommendations for scheduling backups" on page 148

Backing up databases following data migration

At the end of the SQL Server database and transaction log migration process, the Configuration Wizard reminds you to make an immediate backup of the SQL Server databases. Making an immediate backup of the SQL Server databases is critical because any previous non-SnapManager backups will no longer be valid.

Best time to run a SnapManager backup

To minimize the impact of a SnapManager backup on SQL Server client response time, it is best to run the SQL Server database integrity verification of a SnapManager backup operation—the most CPU-intensive part of the backup—during off-peak SQL Server usage hours, or from a remote machine. Typically, off-peak times are between 6:00 p.m and 7:00 a.m.

Note

To avoid degrading the performance of your production SQL Server, run your database verification operations on a remote server.

Frequency of backups

You do not have to perform multiple SnapManager full backups every day, but the more you do, the fewer SQL Server transaction logs need to be played forward at restore time. At a minimum, you should perform one SnapManager full database backup every 24 hours.

Recommendations for scheduling backups

The more often you create SnapManager backups, the fewer SQL Server transaction logs there are to be played forward at restore time, resulting in a faster restore. However, for best results, observe the following recommendations for scheduling backups and verifications:

- Do not schedule any SnapManager operations to overlap each other. Only one SnapManager operation can be running on the same machine at the same time.
- Do not schedule a backup to occur while a database verification is being performed, even if the verification is performed on a remote verification machine. This can result in a backup that cannot be deleted easily. For more information about this problem, see "Busy Snapshot error prevents deletion of backup set" on page 210.
- Do not schedule verifications on the SQL Server server during peak usage hours. The verification process is CPU-intensive and could degrade SQL Server performance if run on the SQL Server during peak usage hours.

One way to conform to the preceding recommendations is to schedule your backups to occur during peak usage hours, and then use the off-peak hours to perform database integrity verifications.

Note

If you are scheduling backups in a Windows cluster, schedule them on only one node. If the SQL Server virtual server fails over to another node, the node that takes over that virtual server performs the backup as scheduled. If the entire node fails, you need to set up the backup schedule again.

Backing Up Databases Using SnapManager 9

This section describes how to back up your SQL Server databases to SnapManager backup sets and how to verify databases stored in those backup sets. The following topics are covered:

- "How SnapManager backup functions are accessed" on page 150
- "Creating a full database backup using SnapManager" on page 152
- "Creating a transaction log backup using SnapManager" on page 172
- "What to do if a SnapManager backup operation fails" on page 184
- "Performing database verification using SnapManager" on page 187
- "Using backup management groups in backup and verification" on page 198
- "Explicitly deleting backup Snapshot copies using SnapManager" on page 202

Attention

You must run SnapManager from the system console, not from a Terminal Services client. Do not use Terminal Services for any type of SnapManager administration, because you might miss critical information that is displayed only in pop-up boxes at the system console.

Related topics:

- Chapter 8, "Understanding SnapManager Backup Sets," on page 129
- Appendix A, "Tools for Managing Backup and Verification," on page 339

How SnapManager backup functions are accessed

To start or schedule a database backup or verification job, you can use either the Backup and Verification tab or the SnapManager Backup wizard to specify the details of the operation you want SnapManager to perform. Depending on the specific parameters you choose, various default SnapManager settings for backup operations and verification settings also come into play.

The following topics introduce the elements of the SnapManager graphical user interface that are used to start or schedule a database backup or verification job:

- "Backup and Verification tab" on page 150
- "SnapManager Backup Wizard" on page 150
- "Default backup settings" on page 151
- "Default verification settings" on page 151

Backup and Verification tab

The SnapManager main window includes a Backup and Verification tab that you can use to specify the job-specific parameters of a SnapManager backup operation or database verification.

This tab can be used to start or schedule the following operations:

- "Full database backup using the Backup and Verification tab" on page 157
- "Transaction log backup using the Backup and Verification tab" on page 175
- "Database verification using the Backup and Verification tab" on page 189

SnapManager Backup Wizard

An alternative to the Backup and Verification tab, the SnapManager Backup Wizard guides you through the specification of the backup or verification operation you want performed.

This wizard can be used to start or schedule the following operations:

- "Full database backup using the Backup Wizard" on page 162
- "Transaction log backup using the Backup Wizard" on page 179
- "Database verification using the Backup Wizard" on page 193

Default backup settings

The Backup Settings dialog box enables you to view or change the default settings that pertain to SnapManager backup operations.

Various default values specified in this dialog box are used when you perform a full database backup, a transaction log only backup, or a database verification in unverified backup sets. This is described in the following topics:

- "Information you need to specify for a full database backup" on page 152
- "Information you need to specify for a transaction log backup" on page 172
- "Information you need to specify for a database verification" on page 187

For more information, see "SnapManager backup options" on page 378.

Default verification settings

The Verification Settings dialog box enables you to view or change the default settings that pertain to the verification of databases in SnapManager backup sets.

Various default values specified in this dialog box are used when you perform a full database backup or when you verify the databases in unverified backup sets. This is described in the following topics:

- "Information you need to specify for a full database backup" on page 152
- "Information you need to specify for a database verification" on page 187

For more information, see "Database integrity verification options" on page 372.

Creating a full database backup using SnapManager

SnapManager provides two ways for you to start or schedule a full database backup: using the Backup Wizard or using the Backup and Verification tab. See the following topics for more information:

- "Information you need to specify for a full database backup" on page 152
- "Full database backup using the Backup and Verification tab" on page 157
- "Full database backup using the Backup Wizard" on page 162
- "Concurrent backup of multiple databases" on page 168

Information you need to specify for a full database backup

A full database backup operation is specified using a combination of parameters.

Job-specific parameters: Each time you start or schedule a full database backup, you must specify the following information in either the Backup and Verification tab or in the Backup Wizard.

- Which databases you want to back up

  When you select one database, SnapManager automatically selects all other databases that are located on the same Data ONTAP storage volume. SnapManager allows you to deselect automatically selected databases as follows:

  - You can deselect databases that reside on a different LUN from the database(s) you selected, even if the LUN is on the same storage volume. If the other LUN stores only a single database, you can deselect or reselect that database individually. If the other LUN houses multiple databases, you must deselect or reselect those databases as a group.
  - In the case of a *stream-based* full database backup, you can deselect any automatically selected databases. However, unless the selected databases share the same one or two LUNs with other databases, SnapManager asks you to confirm your selection; backing up only a subset of the databases that reside on the same volume is not recommended. For more information about the stream-based and online Snapshot backup methods, see "Full database backup" on page 132.

  When you select databases *at the SQL Server instance level* and one of the selected databases cannot be backed up for an unexpected reason (such as the database being offline or in a loading state at the time of the backup), the full database backup operation will progress as follows:

- The *backup report* will include a message at the beginning of the summary section that indicates that the backup was only partially completed because one or more databases were skipped.
- A warning event will be logged to the *event log*. The description field of this event will contain the summary section of the report.
- If *e-mail notification* is enabled, an e-mail notification will be sent to the configured e-mail address.
- *Deletion* of previously created backups will not proceed unless some of the databases in the selected instance are located on local disk at the time the backup is scheduled for execution at a later time.

  If the databases on the LUNs on Data ONTAP storage volumes are moved to local disk later, the previously scheduled backup operation will skip backup deletion. In this case, the backup operation will need to be re-created and rescheduled in order for backup deletion to proceed for future operations.

Note

For an instance-level *transaction-log-only* backup operation in which one of the selected databases cannot be backed up, the operation will proceed in the same manner as described above.

If an SQL Server 2005 database has the full-text search option enabled, the *full-text search catalogs* are visible when you click the "+" next to the database name. The full text catalogs can be migrated, backed up, and restored along with the other files or filegroups of the database.

- Which backup management group you want to assign to this backup

  For details, see "Using backup management groups in backup and verification" on page 198.
- Whether you want to automatically run a transaction log backup after the full database backup is finished
- Whether you want to automatically delete the oldest full database backups within this backup management group (recommended in order to manage the number of Snapshot copies)

  For a description of this option, see "Automatic deletion of the oldest backups in a management group" on page 144.
- If you choose to automatically delete the oldest full database backups within this backup management group: whether you also want to retain up-to-the-minute restore ability for older full database backups in other backup management groups

  For a description of this option, see "Option to retain up-to-the-minute restore ability" on page 146.

- Whether you want to perform a database integrity verification of the Snapshot copy after the full database backup is finished

  Note
  Although it is possible to restore from an unverified backup, NetApp strongly recommends that you restore only from verified backups.

- Whether you want to run a command after the backup is complete (usually done to archive backups)

  This feature is typically used to automatically archive a backup, as described in "Automatically archiving SnapManager backups using SnapVault" on page 278.
- If the volumes you are backing up to are SnapMirror sources: whether you want the destination volumes to be updated after the full database backup is finished

  For more information about this option, see "Replicating Backups to Mirrored Volumes" on page 255.
- Whether you want to run the backup now or schedule it for later

  Note
  If you want to schedule the backup to run later, you also need to know the job scheduling information.

Backup settings: The following list summarizes the backup settings that pertain to full database backups:

- Whether you want the database Snapshot copies and SnapInfo Snapshot copies to be named using generic ("`__recent`") or unique (timestamped) naming conventions

  For more information, see "SnapManager Snapshot naming conventions" on page 142.
- Whether you want to verify the integrity of the live database *before* the backup is performed and whether you want to verify the integrity of the live database *after* the backup is performed

  Note
  Verifying the integrity of the live database is a time-consuming operation. By default, neither of these options is selected.

The preceding options are configured using the Full Database Backup tab of the Backup Settings dialog box, described in "Configuring the profile of a full database backup" on page 378.

The Backup Settings dialog box can be accessed from the Backup and Verification tab, as well as from the Backup Wizard.

- If you are using the Backup and Verification tab, you can open the Backup Settings dialog box from the Options menu.
- From within the Backup Wizard, you can open the Backup Settings dialog box by clicking the Backup Settings button in the Advanced Backup Options screen.

Verification settings: The following list summarizes the settings that pertain to database verification:

- Which SQL Server is used to perform database verification

  This is configured using the SQL Server tab of the Verification Settings dialog box, described in "Selecting the database verification server" on page 372. If you will be specifying a remote verification server, be sure it is set up properly, as described in "Requirements for a remote verification server" on page 29.

  Note

  Database verification is performed on a LUN backed up by Snapshot copy. If you create a Snapshot copy of the same volume while a LUN backed by Snapshot copy exists, you create a "busy Snapshot," which might cause problems when you attempt to delete some Snapshot copies.

  The following precautions will help you to avoid creating a busy Snapshot situation:

  - Do not schedule backups while a verification is in progress.
  - Do not create Snapshot copies at the volume or SnapDrive level.

  For information about busy Snapshot copies, see "Busy Snapshot error prevents deletion of backup set" on page 210.
- Which DBCC options are used to verify database backup Snapshot copies

  This is configured using the DBCC Options tab of the Verification Settings dialog box, described in "Selecting DBCC options" on page 374.

The Verification Settings dialog box can be accessed from the Backup and Verification tab, as well as from the Backup Wizard:

- If you are using the Backup and Verification tab, you can open the Verification Settings dialog box from the Options menu.
- From within the Backup Wizard, you can open the Verification Settings dialog box by clicking Database Verification Settings in the View or Change Database Verification Options screen.

Note

The View or Change Database Verification Options screen appears only if you are specifying a “Full database backup operation” or a “Verify most recent Snapshot copy backups” operation; the SnapManager Backup Wizard does not present this screen if you are specifying a “Transaction log backup only” operation.

Full database backup using the Backup and Verification tab

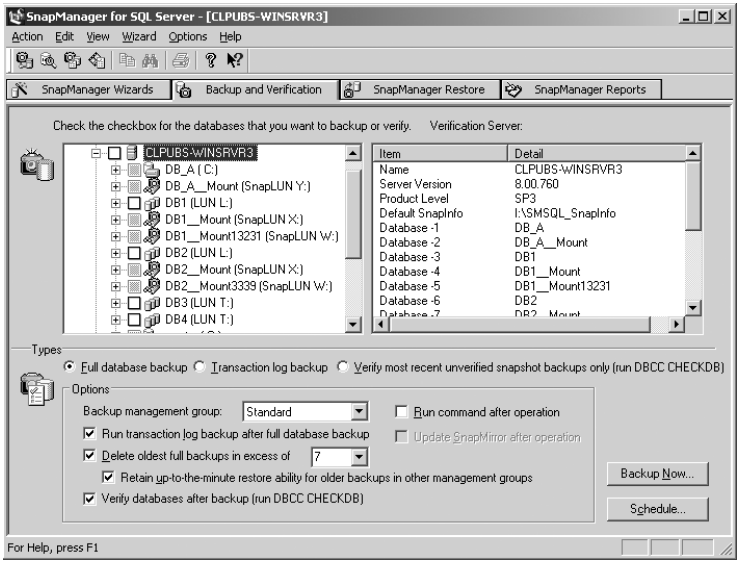
To start or schedule a full database backup using the Backup and Verification tab, complete the following steps.

With a full database backup, you can choose to also back up the associated transaction logs after the database backup is finished. If you want to back up only transaction logs, see "Creating a transaction log backup using SnapManager" on page 172.

Note

For a list of information you need to provide as you complete these steps, see "Information you need to specify for a full database backup" on page 152.

Step	Action
1	In the SnapManager main window, click the Backup and Verification tab. **Result:** The Backup and Verification tab is activated. The *Microsoft SQL Servers* navigation tree in the left panel lists the SQL Server databases that are managed from the current SQL Server. Databases that reside on the same filer volume are shown with disk icons of the same color.
2	In the left panel, select the databases for which you want to perform a full backup. When you select a database, SnapManager automatically selects all other databases that reside on the same filer volume. For information about deselecting any automatically selected databases, see the bullet "Which databases you want to back up" under "Job-specific parameters" on page 152. When you select databases at the *SQL Server instance level*, SnapManager will report any offline databases as skipped in the backup report. For more information, see "Information you need to specify for a full database backup" on page 152.

Step	Action
3	In the Types panel, select the Full database backup option. **Result:** The Options subpanel provides a choice of backup options that pertain to full database backup operations. 
4	In the Backup Management Group option, select the management group for the backup you want to create: Standard, Daily, or Weekly. For more information, see “SnapManager Snapshot naming conventions” on page 142 and “Using backup management groups in backup and verification” on page 198.
5	If you want the database backup operation to be immediately followed by a transaction log backup, select the “Run transaction log backup after full database backup” option.

Step	Action
6	If you want to automatically delete older backups within the selected backup management group, do the following: **a.** Select the Delete Oldest Backups In Excess Of option. **b.** Select the number of full database backups you want to keep. Selecting this option enables you to manage the number of Snapshot copies for a given backup management group. For more information, see "Automatic deletion of the oldest backups in a management group" on page 144.
7	If you choose to automatically delete older backups within a backup management group, you are also given the option to ensure that this is done without creating a break in the continuity of transaction logs between the previous backup and the present time. By limiting the automatic deletion of backups in this way, you retain the ability to use the older backups in an up-to-the-minute restore. To use this option, select the "Retain up-to-the-minute restore ability for older backups in other management groups" option. For more information, see "Automatic deletion of the oldest backups in a management group" on page 144.
8	If you want the full database backup to be followed by a database verification, select the "Verify Snapshot after backup (run DBCC)" option. Otherwise, be sure to verify the backup later, as described in Chapter 9, "Performing database verification using SnapManager," on page 187. **Note** Although it is possible to restore from an unverified backup, NetApp strongly recommends that you restore only from verified backups.

Step	Action
9	If you want to run a command after the backup finishes, select the Run Command After Operation option. This is usually done to archive backups. **Note** If you select this option, SnapManager will prompt you for the details when you are ready to start or schedule the backup operation.
10	If the databases reside on a SnapMirror source volume and you want the destination volume to be updated after this backup finishes, select the Update SnapMirror After Operation option. **Note** If you select the Update SnapMirror After Operation option and any of the selected database files or associated SnapInfo directories are hosted on LUNs that are not configured for SnapMirror replication, only the databases housed on mirrored LUNs will be updated on the SnapMirror destination volumes.
Backup now or schedule for later?	
11	You can either run the backup now or schedule it for later. Click one of the following buttons: ◆ Backup Now ◆ Schedule
12	If the Run Command After Operation dialog box appears, specify the command and then Click OK to close the dialog box. For details, see "Using the Run Command After Operation dialog box" on page 345.
To complete this procedure by scheduling the backup:	
13	If the Schedule Job dialog box appears, complete this procedure by scheduling the backup. For details, see "Scheduling a backup job or a database verification job" on page 349.

Step	Action
To complete this procedure by starting the backup:	
14	If the Backup Status dialog box appears, do the following: **a.** Read the items displayed in the Backup Task List. This list will be used to show the progress of the backup operation after you start it. **b.** When you are ready to start the backup operation, click Start Now. **Result:** The backup operation is performed, and each item in the Backup Task List is checked off as the task is finished. ◆ You can toggle the Backup Status dialog box between two different views—Backup Task List view and Backup Report view—by using the Switch button on either view. ◆ The Backup Report view displays detailed progress information as the backup progresses. You can also print this information by using the Print Report button. ◆ If the backup is successful, the Backup Task List view shows the check-off list with the tasks completed. ◆ If Notification is enabled, e-mail is sent and the event is posted to the Windows Application event log. **Note** A SnapManager backup operation encompassing multiple databases can fail if the SQL server is not configured with a sufficient number of worker threads to support this operation. For more information, see "Concurrent backup of multiple databases" on page 168.

Full database backup using the Backup Wizard

To start or schedule a full database backup using the Backup Wizard, complete the following steps.

Note

For a list of information you need to provide as you complete these steps, see "Information you need to specify for a full database backup" on page 152.

Step	Action
1	In the SnapManager for SQL Server main window, click the SnapManager Backup Wizard icon. **Result:** The SnapManager Backup Wizard starts and displays the Welcome screen.
Welcome	
2	Click Next. **Result:** The Databases to Back Up or Verify screen appears. The *Microsoft SQL Servers* navigation tree in the left panel lists the SQL Server databases that are managed from the current SQL Server. Databases that reside on the same filer volume are shown with disk icons of the same color.
Databases to Back Up or Verify	
3	In the left panel, click to select the databases you want to back up. When you select a database, SnapManager automatically selects all other databases that reside on the same storage system volume. For information about deselecting any automatically selected databases, see the bullet "Which databases you want to back up" under "Job-specific parameters" on page 152. When you select databases at the *SQL Server instance level*, SnapManager will list any offline databases as skipped in the backup report. For more information, see "Information you need to specify for a full database backup" on page 152.
4	Click Next. **Result:** The Back Up or Verify Databases and Transaction Logs screen appears.

Step	Action
Back Up or Verify Databases and Transaction Logs	
5	Select the Back Up Databases and Transaction Logs option, and then click Next. **Result:** The Backup Type screen appears.
Backup Type	
6	Select the Full Database Backup option, and then click Next. **Result:** The Backup Management Group screen appears.

Step	Action
Backup Management Group	
7	Follow the instructions in the remainder of the Backup Wizard screens. The following screens enable you to specify the details of a full database backup: ◆ Backup Management Group ◆ Back Up Transaction Log After Full Database Backup ◆ Delete the Oldest Full Backups ◆ Retain Up-to-the-Minute Restore Ability for Older Backups **Note** This screen appears only if you selected the Delete the Oldest Backups option in the previous screen. ◆ Verify the Databases in this Backup **Note** To view or change the verification settings, click Verification Settings. ◆ View or Change Verification Settings **Note** This screen appears only if you chose to automatically verify the databases as soon as the backup is created. ◆ Advanced Backup Options ◆ Option to Perform SnapMirror Update After Operation **Note** This screen appears only if any SQL Server databases or SnapInfo directories reside on a SnapMirror source volume. ◆ Run a Command After the Operation For a list of information you need to provide, see “Information you need to specify for a full database backup” on page 152.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">When to Run this Operation</td></tr>
<tr><td>8</td><td>The When to Run this Operation screen prompts you to choose whether you want the operation to be performed now or scheduled for a later time.<table><tr><th>If you want to...</th><th>Then...</th></tr><tr><td>Run the backup now</td><td>Go to Step 9.</td></tr><tr><td>Schedule the backup for later</td><td>Go to Step 12.</td></tr></table></td></tr>
<tr><td colspan="2">To run the backup now:</td></tr>
<tr><td>9</td><td>If you want to run the backup now, do the following:
a. Leave the Now option selected.
b. Click Next.
Result: The Completing the Backup Wizard screen appears and displays the backup settings you have selected.</td></tr>
<tr><td>10</td><td>After you verify that all the settings in the window are correct, go to the Completing the Backup Wizard dialog box and click Finish.
Result: The Backup Wizard closes and the Backup Status window appears and displays a Backup Task List that will be used to show the progress of the backup operation after you start it.</td></tr>
</table>

Step	Action
11	In the Backup Status window, click Start Now. **Result:** The backup is performed and the Snapshot copy is written to the volume. ◆ The Backup Status dialog box appears and displays a Backup Task List that will be used to show the progress of the backup operation after you start it. ◆ SnapManager Backup completes each task and checks it off on the list shown in the Backup Task List view. ◆ You can switch back and forth between the task check-off list and the progress report using the Switch buttons on either view. ◆ If the backup is successful, the Task window shows the check-off list with the tasks completed. ◆ If Notification is enabled, e-mail is sent and the event is posted to the Windows Application event log. **Note** A SnapManager backup operation encompassing multiple databases can fail if the SQL server is not configured with a sufficient number of worker threads to support this operation. For more information, see "Concurrent backup of multiple databases" on page 168.
To schedule the backup for later:	
12	If you want to schedule the backup for later, do the following: **a.** Select the Later option. **b.** In the Job Name box, enter a name for this job. **c.** If you want this job name to overwrite a job of the same name (if it exists), select the Replace Job if Exists option. **d.** Click Next. For more information, see "Scheduling a backup job or a database verification job" on page 349. **Result:** The Select the Scheduling Service screen appears.

Step	Action
13	In the Select the Scheduling Service screen, do the following: **a.** Select either SQL Server Agent or Windows Scheduled Tasks. For more information, see "Scheduling a backup job or a database verification job" on page 349. **b.** Click Next. **Result:** The Completing the Backup Wizard screen appears and displays the backup settings you have selected.
14	After you verify that all the settings in the window are correct, go to the Completing the Backup Wizard dialog box and click Finish. **Result:** The selected scheduler displays a dialog box. ◆ If you chose to schedule the backup job using the SQL Server Agent, the Properties dialog box appears. ◆ If you chose to schedule the backup job using Scheduled Tasks, the Schedule Job dialog box appears.
15	To schedule the job and close the Backup Wizard, do the following: **a.** Specify the details of the job. **b.** Click OK.

Concurrent backup of multiple databases

When attempting to back up multiple databases concurrently, the operation might fail due to lack of a sufficient number of worker threads.

Symptom: The backup failure is accompanied by one of the following error messages:

- `VDI backup threads backup preparation failed.`
- `Error code 0xc00408d1`

  `The number of databases selected for concurrent backup exceeds the maximum number of SQL server worker threads that are available on the SQL server` *ServerName* `to perform this backup. Please increase the number of worker threads for SQL server` *ServerName* `to be greater than the sum of the worker thread required to service SQL client and the number of databases connected to SQL server` *ServerName*`, which are selected for concurrent backup. Using Enterprise Manager, the maximum number of worker threads can be configured from the Processor tab in the properties window for the SQL server` *ServerName*`.`

Workaround 1: Decrease the number of databases being backed up concurrently.

Workaround 2: Increase the maximum number of worker threads for this SQL Server.

Workaround procedure: To implement either or both of the preceding workarounds, complete the following steps.

Step	Action
1	Start the Configuration Wizard, then step through the following screens without specifying any configuration changes: ◆ Database Verification Server and Account Authentication ◆ Where to Move Databases

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Number of Databases that Can be Backed Up Concurrently</td></tr>
<tr><td>2</td><td>If you want to decrease the maximum number of databases that can be backed up concurrently for each SQL Server instance (as described in Workaround 1), decrease the value in the “Maximum databases...” box in the Number of Databases that Can be Backed Up Concurrently screen.

For more detailed information, see “Using the Number of Databases that Can Be Backed Up Concurrently screen” on page 417.</td></tr>
<tr><td>3</td><td>If you want to increase the maximum number of SQL Server worker threads (as described in Workaround 2), select the Modify SQL Server “Maximum Worker Threads Parameter” option in the Number of Databases that Can be Backed Up Concurrently screen.

For more detailed information, see “Using the Number of Databases that Can Be Backed Up Concurrently screen” on page 417.</td></tr>
<tr><td>4</td><td>Click Next.

The next screen that appears depends on whether you selected the Modify SQL Server “Maximum Worker Threads” Parameter option.

<table><tr><th>If...</th><th>Then...</th></tr><tr><td>You did not select the Modify SQL Server “Maximum Worker Threads” Parameter option</td><td>The SnapInfo Directory Type screen appears. Go to Step 7.</td></tr><tr><td>You selected the Modify SQL Server “Maximum Worker Threads” Parameter option</td><td>The Modify SQL Server Maximum Worker Threads Parameter screen appears. Go to Step 5.</td></tr></table></td></tr>
</table>

Step	Action
Modify SQL Server Maximum Worker Threads Parameter	
5	Use the Modify SQL Server Maximum Worker Threads Parameter screen to specify whether you want to modify the maximum number of worker threads for any SQL Server instance. You have the following options: ◆ Leave the current maximum number of worker threads unchanged for all SQL Server instances ◆ Apply all of the recommended new values displayed in the New Max Worker Threads column ◆ Edit one or more of the recommended new values before applying the changes For more detailed information, see "Using the Modify SQL Server Maximum Worker Threads Parameter screen" on page 419. **Note** If you choose to change the maximum number of worker threads for any SQL Server instances, those instances will be restarted when you click Finish in the Completing the Configuration Wizard screen.
6	After you have finished viewing and optionally changing this SQL Server parameter, click Next. **Result:** The SnapInfo Directory Type screen appears.

Step	Action
Complete the Configuration Wizard	
7	Step through the remaining screens of the Configuration Wizard without specifying any further configuration changes: ◆ SnapInfo Directory Type ◆ Either of the following sequence of screens, as applicable: ❖ Specify a Single SnapInfo Directory ❖ Where to Put the Default SnapInfo Directory for Each SQL Server Instance Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases Where to Put the SnapInfo Directory for Each Database ◆ Database Migration Options ◆ Add Microsoft iSCSI Service Dependency ◆ Configure Automatic Event Notification
8	In the Completing the Configuration Wizard screen, verify the changes you specified and then click Finish.
9	In the Configurator Status dialog box, click Start Now to close the dialog box and begin the Configuration Wizard tasks you specified.
10	When a message box appears and notifies you that the configuration changes completed successfully, click OK to close the message box.
11	Click Close to close the Configurator Status dialog box.

Creating a transaction log backup using SnapManager

SnapManager provides two ways for you to start or schedule a transaction log backup: using the SnapManager Backup Wizard or using the Backup and Verification tab. See the following topics for more information:

- "Information you need to specify for a transaction log backup" on page 172
- "Transaction log backup using the Backup and Verification tab" on page 175
- "Transaction log backup using the Backup Wizard" on page 179

Note

The topics in this section describe how to start or schedule a SnapManager backup of SQL Server transaction logs only. If you want to back up transaction logs as a follow-on task to a successful full database backup, see "Creating a full database backup using SnapManager" on page 152.

Information you need to specify for a transaction log backup

A backup of only transaction logs is specified using a combination of parameters.

Job-specific parameters: Each time you start or schedule a transaction log only backup, you must specify the following information in either the Backup and Verification tab or in the Backup Wizard:

- Which databases you want to back up

  When you select one database, SnapManager automatically selects all other databases that are located on the same Data ONTAP storage volume. You can deselect databases that reside on a different LUN from the database(s) you selected, even if the LUN is on the same storage volume.

  - If the other LUN stores only a single database, you can deselect or reselect that database individually.
  - If the other LUN houses multiple databases, you must deselect or reselect those databases as a group.

  When you select databases *at the SQL Server instance level* and one of the selected databases cannot be backed up for an unexpected reason (such as the database being offline or in a loading state at the time of the backup), the transaction-log-only backup operation will progress as follows:

  - The *backup report* will include a message at the beginning of the summary section that indicates that the backup was only partially completed because one or more databases were skipped.
  - A warning event will be logged to the *event log*. The description field of this event will contain the summary section of the report.

- If *e-mail notification* is enabled, an e-mail notification will be sent to the configured e-mail address.
- *Deletion* of previously created backups will not proceed unless some of the databases in the selected instance are located on local disk at the time the backup is scheduled for execution.

  If the databases on the LUNs on Data ONTAP storage volumes are moved to local disk at later time, the previously scheduled backup operation will skip backup deletion. In this case, the backup operation will need to be re-created and rescheduled in order for backup deletion to proceed for future operations.

Note

For an instance-level *full database* backup operation in which one of the selected databases cannot be backed up, the operation will proceed in the same manner as described above.

- Whether you want to automatically delete the oldest transaction log backups (recommended in order to manage the disk space occupied by the SnapInfo directory)

  For a description of this option, see "Automatic deletion of the oldest backups in a management group" on page 144.
- Whether you want to run a command after the backup is complete (usually done to archive backups)

  This feature is typically used to archive the backup, as described in "Automatically archiving SnapManager backups using SnapVault" on page 278.
- If the transaction logs you are backing up are for databases and related SnapInfo directories that are located on SnapMirror sources: whether you want the destination volumes to be updated after the transaction log backup is finished

  For more information about this option, see "Replicating Backups to Mirrored Volumes" on page 255.
- Whether you want to run the backup now or schedule it for later

Note

If you want to schedule the backup to run later, you also need to know the job scheduling information.

Backup settings: The following list summarizes the backup settings that pertain to transaction log only backups. These settings enable tasks that can be performed after the transaction log backup finishes successfully.

- Whether you want to create a Snapshot copy of the LUN that contains the SnapInfo directory.
- If you create a Snapshot copy of the SnapInfo directory after the backup is finished: whether you also want to delete the oldest SnapInfo directory Snapshot copies and retain only a certain number of the most recent Snapshot copies.

  Selecting this option enables you to manage the number of SnapInfo Snapshot copies.
- Whether you want to truncate the transaction log itself by removing any inactive entries (transaction already committed to the database)

  Selecting this option enables you to manage the size of the transaction log.

The preceding options are configured using the Transaction Log Backup tab of the Backup Settings dialog box, described in "Configuring the profile of a transaction log backup" on page 380.

The Backup Settings dialog box can be accessed from the Backup and Verification tab, as well as from the Backup Wizard.

- From within the Backup Wizard, you can open the Backup Settings dialog box by clicking the Backup Settings button in the Advanced Backup Options screen.
- If you are using the Backup and Verification tab, you can open the Backup Settings dialog box from the Options menu.

For more details, see "Configuring the profile of a transaction log backup" on page 380.

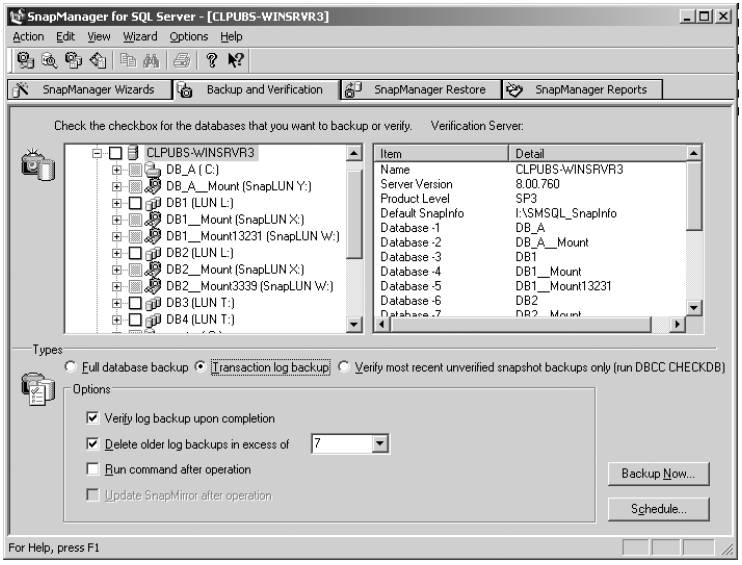
Transaction log backup using the Backup and Verification tab

To start or schedule a backup of only transaction logs using the Backup and Verification tab, complete the following steps. If you want to automatically back up transaction logs after a full database backup is finished, see "Creating a full database backup using SnapManager" on page 152 instead.

Note

For a list of information you need to provide as you complete these steps, see "Information you need to specify for a transaction log backup" on page 172.

Step	Action
1	In the SnapManager main window, click the Backup and Verification tab.
2	In the left panel, select the databases you want to back up. When you select a database, SnapManager automatically selects all other databases that reside on the same storage system volume. For information about deselecting any automatically selected databases, see the bullet "Which databases you want to back up" under "Job-specific parameters" on page 172. When you select databases at the SQL Server instance level, SnapManager will report any offline databases as skipped in the backup report. For more information, see "Information you need to specify for a transaction log backup" on page 172.

Step	Action
3	In the Types panel, select the Transaction log backup option. **Result:** The Options subpanel provides a choice of backup options that pertain to backing up only transaction logs. 
4	If you want the transaction log backup to be followed by a verification, select the “Verify Log Backup upon Completion” option.
5	If you want to automatically delete older transaction log backups, select the “Delete Oldest Backups In Excess Of” option and select the number of transaction log backups you want to keep. Selecting this option enables you to manage the number transaction log backups.
6	If you want to run a command after the backup finishes, select the Run Command After Operation option. This is usually done to archive backups. **Note** If you select this option, SnapManager will prompt you for the details when you are ready to start or schedule the backup operation.

Step	Action
Backup now or schedule for later?	
7	You can either run the backup now or schedule it for later. Click one of the following buttons: ◆ Backup Now ◆ Schedule
8	If the Run Command After Operation dialog box appears, specify the command, and then Click OK to close the dialog box. For details, see "Using the Run Command After Operation dialog box" on page 345.
To complete this procedure by scheduling the backup:	
9	If the Schedule Job dialog box appears, complete this procedure by scheduling the backup. For details, see "Scheduling a backup job or a database verification job" on page 349.

Step	Action
To complete this procedure by starting the backup:	
10	If the Backup Status dialog box appears, do the following: **a.** Read the items displayed in the Backup Task List. This list will be used to show the progress of the backup operation after you start it. **b.** When you are ready to start the backup operation, click Start Now. **Result:** The backup operation is performed, and each item in the Backup Task List is checked off as the task is finished. ◆ You can toggle the Backup Status dialog box between two different views—Backup Task List view and Backup Report view—by using the Switch button on either view. ◆ The Backup Report view displays detailed progress information as the backup progresses. You can also print this information by using the Print Report button. ◆ If the backup is successful, the Backup Task List view shows the check-off list with the tasks completed. ◆ If Notification is enabled, e-mail is sent and the event is posted to the Windows Application event log.

Transaction log backup using the Backup Wizard

To start or schedule a backup of only transaction logs using the Backup Wizard, complete the following steps.

Note

For a list of information you need to provide as you complete these steps, see "Information you need to specify for a transaction log backup" on page 172.

Step	Action
1	In the SnapManager for SQL Server main window, click the SnapManager Backup Wizard icon. **Result:** The SnapManager Backup Wizard starts and displays the Welcome screen.
Welcome	
2	Click Next. **Result:** The Databases to Back Up or Verify screen appears.
Databases to Back Up or Verify	
3	In the left panel, click to select the databases you want to back up, then click Next. When you select a database, SnapManager automatically selects all other databases that reside on the same storage system volume. For information about deselecting any automatically selected databases, see the bullet "Which databases you want to back up" under "Job-specific parameters" on page 172. When you select databases at the SQL Server instance level, SnapManager will report any offline databases as skipped in the backup report. For more information, see "Information you need to specify for a transaction log backup" on page 172. **Result:** The Back Up or Verify Databases and Transaction Logs screen appears.

Step	Action
Back Up or Verify Databases and Transaction Logs	
4	Select the Back Up Databases and Transaction logs option, and then click Next. **Result:** The Backup Type screen appears.
Backup Type	
5	Select the Transaction Log Backup Only option, and then click Next. **Result:** The Delete the Oldest Transaction Log Backups screen appears.
Delete the Oldest Transaction Log Backups	
6	Follow the instructions in the remainder of the Backup Wizard screens. The following screens enable you to specify the details of a transaction log backup only: ◆ Delete the Oldest Transaction Log Backups ◆ Verify the Transaction Logs in this Backup **Note** To view and change the verification settings, click Verification Settings. ◆ Advanced Backup Options ◆ Option to Perform SnapMirror Update After Operation **Note** This screen appears only if any SQL Server databases or SnapInfo directories reside on a SnapMirror source volume. ◆ Run a Command After the Operation For a list of information you need to provide, see “Information you need to specify for a transaction log backup” on page 172.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">When to Run this Operation</td></tr>
<tr><td>7</td><td>The When to Run this Operation screen prompts you to choose whether you want the operation to be performed now or scheduled for a later time.<table><tr><th>If you want to...</th><th>Then...</th></tr><tr><td>Run the backup now</td><td>Go to Step 8.</td></tr><tr><td>Schedule the backup for later</td><td>Go to Step 11.</td></tr></table></td></tr>
<tr><td colspan="2">To run the backup now:</td></tr>
<tr><td>8</td><td>If you want to run the backup now, do the following:
a. Leave the Now option selected.
b. Click Next.
Result: The Completing the Backup Wizard screen appears and displays the backup settings you have selected.</td></tr>
<tr><td>9</td><td>After you verify that all the settings in the screen are correct, click Finish.
Result: The Backup Wizard closes and the Backup Status window appears and displays a Backup Task List that will be used to show the progress of the backup operation after you start it.</td></tr>
</table>

Step	Action
10	In the Backup Status window, click Start Now. **Result:** The backup is performed and the Snapshot copy is written to the volume. ◆ The Backup Status dialog box appears and displays a Backup Task List that will be used to show the progress of the backup operation after you start it. ◆ SnapManager Backup completes each task and checks it off on the list shown in the Backup Task List view. ◆ You can switch back and forth between the task check-off list and the progress report using the Switch buttons on either view. ◆ If the backup is successful, the Task window shows the check-off list with the tasks completed. ◆ If Notification is enabled, e-mail is sent and the event is posted to the Windows Application event log.
To schedule the backup for later:	
11	If you want to schedule the backup for later, do the following: **a.** Select the Later option. **b.** In the Job Name box, enter a name for this job. **c.** If you want this job name to overwrite a job of the same name (if it exists), select the Replace Job if Exists option. **d.** Click Next. For more information, see "Scheduling a backup job or a database verification job" on page 349. **Result:** The Select the Scheduling Service screen appears.

Step	Action
12	In the Select the Scheduling Service screen, do the following: **a.** Choose either SQL Server Agent or Windows Scheduled Tasks. For more information, see "Scheduling a backup job or a database verification job" on page 349. **b.** Click Next. **Result:** The Completing the Backup Wizard screen appears and displays the backup settings you have selected.
13	After you verify that all the settings in the window are correct, go to the Completing the Backup Wizard dialog box and click Finish. **Result:** The selected scheduler displays a dialog box. ◆ If you chose to schedule the backup job using the SQL Server Agent, the Properties dialog box appears. ◆ If you chose to schedule the backup job using Windows Scheduled Tasks, the Schedule Job dialog box appears.
14	To schedule the job and close the Backup Wizard, do the following: **a.** Specify the details of the job. **b.** Click OK.

What to do if a SnapManager backup operation fails

If a SnapManager backup fails, check the backup report for details about what SnapManager was trying to do when the failure occurred. SnapManager reports are described in Chapter 11, "Managing SnapManager Operational Reports," on page 249. You can also review the following list of common backup failures:

- "Snapshot limit reached" on page 184
- "SnapInfo directory being accessed" on page 185
- "SnapInfo directory out of space" on page 185
- "Data does not match" on page 185
- "Snapshot copy already exists" on page 185
- "SnapManager server initialization failed" on page 186
- "A database "not in valid configuration" was not backed up" on page 186
- "[DBMSLPCN] ConnectionRead (WrapperRead())" on page 186

Snapshot limit reached

If any of the volumes containing the LUNs being backed up reach the Snapshot limit of 255—whether those Snapshot copies were created by SnapManager or not—SnapManager backup fails. For more information about the Snapshot copy limit, see "Maximum number of Snapshot copies per volume" on page 144.

Backup Snapshot copies can be automatically deleted: To avoid reaching the per-volume Snapshot copy limit in the future, you can automatically delete the oldest backup Snapshot copies in a backup management group. Automatic backup deletion is performed only after a backup finishes successfully. For more information, see "Automatic deletion of the oldest backups in a management group" on page 144.

Backup Snapshot copies can be explicitly deleted: Backup Snapshot copies can also be explicitly selected for deletion. For a description of the Snapshot copy naming convention, which helps identify the Snapshot copy contents, see "SnapManager Snapshot naming conventions" on page 142. For more information, see "Explicitly deleting backup Snapshot copies using SnapManager" on page 202.

Restore Snapshot copies can be explicitly deleted: For a description of *restore Snapshot copies*, see "How SnapManager Restore works" on page 219. For a description of how to view and delete these Snapshot copies, see "Deleting restored Snapshot copies" on page 247. SnapManager displays restore Snapshot copies only in the Delete Backups dialog box when the Delete Snapshot of LUNs Created During Restore option is selected.

SnapInfo directory being accessed

Because a SnapManager backup might include renaming a SnapInfo subdirectory and Windows does not allow a directory name to be changed while it is being accessed, accessing the SnapInfo directory with a tool such as Windows Explorer could cause the backup to fail. Make sure that you do not hold any exclusive access to the SnapInfo directory on the SQL Server host system while a backup is progress.

SnapInfo directory out of space

Expand the LUN that contains the SnapInfo directory.

Note

When you expand a LUN, make sure that sufficient space remains in the volume for Snapshot copy creation, so that SnapManager can continue to function correctly. For more information about volume sizing, see "Assessing volume size" on page 96.

Data does not match

This error occurs if you made changes to your SQL Server database configuration after SnapManager was started. Any of the following actions will refresh your SnapManager view:

- Press F5 on your keyboard.
- From the SnapManager main window, select Action > Refresh SQL Server Configuration.
- Restart SnapManager.

Snapshot copy already exists

Either of the following circumstances might cause this error to occur:

- The system clock on the host running SnapManager might not be synchronized with the clock on the storage system. These two clocks must be synchronized in order for SnapDrive to function correctly. For more information, see the *Data ONTAP Software Setup Guide* for your version of Data ONTAP.
- If a SnapMirror replication is running when you attempt to initiate a SnapManager backup, the backup can fail. You can avoid this problem by making sure that any SnapMirror replications have enough time to finish before you initiate another SnapManager backup.

SnapManager server initialization failed

Either of the following circumstances might cause this error to occur:

- You have exited the SnapManager application, but one or both of the following processes are still running:
  - `SnapMgrSql.exe`
  - `SmSqlSrvr.exe`

  To correct this problem, use Windows Task Manager to terminate any orphaned SnapManager processes.
- The permissions associated with the SnapManager service account, or the service account itself, have been changed. In this case, SnapManager might not function correctly.

  For information about the SnapManager service account requirements, see "SnapManager service account requirements" on page 26.

A database "not in valid configuration" was not backed up

If a SnapManager backup operation attempts to back up an SQL Server database for which the Auto Shrink option is enabled, the backup operation might fail with the following message in the backup report:

```
WARNING: Database DatabaseName of ServerName is not in valid
configuration, and will not be included in this backup.
```

To avoid this problem, do not enable the Auto Shrink option for SQL Server databases that you back up using SnapManager.

[DBMSLPCN] ConnectionRead (WrapperRead())

If the SnapManager host system is running SQL Server 2005, a SnapManager backup operation might fail with the following message in the backup report:

```
[Microsoft][ODBC SQL Server Driver][DBMSLPCN]ConnectionRead
(WrapperRead()).
```

To avoid this problem, install MDAC 2.8 SP1 on the Windows host. See "Windows host system requirements" on page 23.

Performing database verification using SnapManager

If you created a full database backup Snapshot copy without automatically verifying the databases at the time the backup was created, you can verify the databases in that backup as a separate operation. See the following topics for more information:

- "Information you need to specify for a database verification" on page 187
- "Database verification using the Backup and Verification tab" on page 189
- "Database verification using the Backup Wizard" on page 193

Attention

If you attempt to perform a database verification (or a backup with database verification) when SnapManager is running from a Terminal Services client instead of from a system console, the operation will fail.

Information you need to specify for a database verification

A database integrity verification job is specified using a combination of parameters.

Job-specific parameters: Each time you start or schedule a database verification, you must specify the following information in either the Backup Wizard or in the Backup and Verification tab:

- The databases for which you want to verify any unverified backup Snapshot copies when you select one database, SnapManager automatically selects all other databases that are located on the same Data ONTAP storage volume.

  You can deselect databases that reside on a different LUN from the database(s) you selected, even if the LUN is on the same storage volume, as follows:

  - If the other LUN stores only a single database, you can deselect or reselect that database individually.
  - If the other LUN houses multiple databases, you must deselect or reselect those databases as a group.
- Within the selected databases, the backup management groups for which you want to verify any unverified backup Snapshot copies

  For more information, see "Using backup management groups in backup and verification" on page 198.
- For the selected databases and backup management groups, the number of unverified backup Snapshot copies you want to verify

Note

If you request verification of a greater number of unverified Snapshot copies than specified by your database and backup management group selections, the verification will proceed, and therefore all backups will be verified.

- Whether you want to run a command after the backup is complete (usually done to archive backups)

  This feature is typically used to automatically archive a backup, as described in "Automatically archiving SnapManager backups using SnapVault" on page 278.

- If the volumes you are backing up to are SnapMirror sources: whether you want the destination volumes to be updated after the database verification is finished

  Note

  The result of the database verification operation (the database integrity status) is written to the backup set, in the SnapInfo directory. By replicating the backup set to the mirrored volume, this status information is kept current on the mirrored volume.

  For more information, see "Replicating Backups to Mirrored Volumes" on page 255.

- Whether you want to run the verification now or schedule it for later

  Note

  If you want to schedule the verification to run later, you also need to know the job scheduling information.

Verification settings: The following list summarizes the settings that pertain to database verification:

- Which SQL Server is used to perform database verification

  This is configured using the SQL Server tab of the Verification Settings dialog box, described in "Selecting the database verification server" on page 372. If you will be specifying a remote verification server, be sure it is set up properly, as described in "Requirements for a remote verification server" on page 29.

Note

Whether you run database verification either remotely or locally, the verification is performed on a LUN backed by Snapshot copy. If you make a Snapshot copy of the same volume while a LUN backed by Snapshot copy exists, you create a "busy Snapshot," which might cause problems when you attempt to delete some Snapshot copies. For this reason, you must be careful not to schedule backups while a verification is in progress. For information about busy Snapshot copies, see "Busy Snapshot error prevents deletion of backup set" on page 210.

- Which DBCC options are used to verify database backup Snapshot copies

  This is configured using the DBCC Options tab of the Verification Settings dialog box, described in "Selecting DBCC options" on page 374.

The Verification Settings dialog box can be accessed from the Backup and Verification tab, as well as from the Backup Wizard.

- If you are using the Backup and Verification tab, you can open the Verification Settings dialog box from the Options menu.
- From within the Backup Wizard, you can open the Verification Settings dialog box by clicking Database Verification Settings in the View or Change Database Verification Options screen.

Note

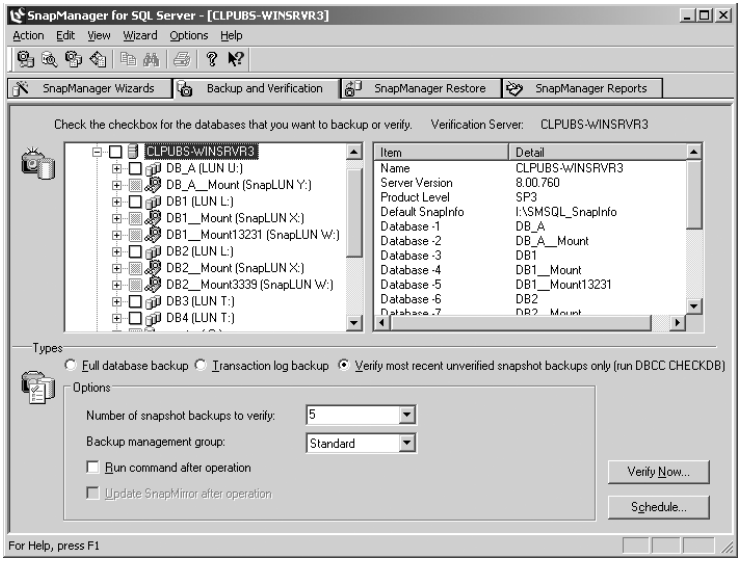
The View or Change Database Verification Options screen appears only if you are specifying a "Full database backup" operation or a "Verify most recent Snapshot backups" operation; the SnapManager Backup Wizard does not present this screen if you are specifying a "Transaction log backup only" operation.

Database verification using the Backup and Verification tab

To start or schedule database verification, complete the following steps *from the production SQL Server host system* (not from the remote verification server).

Note

Whether you run verification remotely or locally, the verification is performed on a LUN backed by Snapshot copy. If you make a Snapshot copy of the same volume while a LUN backed by Snapshot copy exists, you create a "busy Snapshot," which might cause problems when you attempt to delete some Snapshot copies. For this reason, you must be careful not to schedule backups while a verification is in progress. For information about busy Snapshots, see "Busy Snapshot error prevents deletion of backup set" on page 210.

Step	Action
1	On the production SQL server, click the Backup and Verification tab.
2	In the left panel, select the databases you want to verify. When you select a database, SnapManager automatically selects all other databases that reside on the same storage system volume. For information about deselecting any automatically selected databases, see the bullet "Which databases you want to back up" under "Job-specific parameters" on page 187.
3	In the Types panel, select the Verify Databases and Transaction Logs in the Most Recent Unverified Backups (Run DBCC CHECKDB) option. **Result:** The Options subpanel provides a choice of backup options that pertain verifying the unverified backup Snapshot copies. 

Step	Action
4	Select the number of the most recent unverified backups you want to verify. **Note** Only unverified backups are counted. For example, if you select **2**, and all of the databases contained in the most recent backup have already been verified, then SnapManager verifies the databases in the two previous backups.
5	In the Backup Management Group option, select the backup management group of the backups you want to verify. If you want to verify the most recent backups regardless of their backup management group, select All.
6	If you want to run a command after the database verification finishes, select the Run Command After Operation option. This is usually done to archive backups. **Note** If you select this option, SnapManager will prompt you for the details when you are ready to start or schedule the verification operation.
7	If your volume is a SnapMirror source volume and you do not want the destination volume to be updated after this verification is complete, clear the Update SnapMirror After Operation option. **Note** The result of the database verification operation (the database integrity status) is written to the backup set, in the SnapInfo directory. By replicating the backup set to the mirrored volume, this status information is kept current on the mirrored volume.
Verify now or schedule for later?	
8	You can either run the verification now or schedule it for later. Click one of the following buttons: ◆ Verify Now ◆ Schedule

Step	Action
9	If the Run Command After Operation dialog box appears, specify the command and then Click OK to close the dialog box. For details, see "Using the Run Command After Operation dialog box" on page 345.
To complete this procedure by scheduling the verification:	
10	If the Schedule Job dialog box appears, complete this procedure by scheduling the backup. For details, see "Scheduling a backup job or a database verification job" on page 349.
To complete this procedure by starting the verification:	
11	If the Backup Status dialog box appears, do the following: **a.** Read the items displayed in the Backup Task List. This list will be used to show the progress of the verification operation after you start it. **b.** When you are ready to start the verification operation, click Start Now. **Result:** The verification operation is performed, and each item in the Backup Task List is checked off as the task is finished. ◆ You can toggle the Backup Status dialog box between two different views—Backup Task List view and Backup Report view—by using the Switch button on either view. ◆ The Backup Report view displays detailed progress information as the verification progresses. You can also print this information by using the Print Report button. ◆ If the backup is successful, the Backup Task List view shows the check-off list with the tasks completed. ◆ If Notification is enabled, e-mail is sent and the event is posted to the Windows Application event log.

Database verification using the Backup Wizard

To start or schedule a database verification using the Backup Wizard, complete the following steps.

Note

For a list of information you need to provide as you complete these steps, see "Information you need to specify for a database verification" on page 187.

Step	Action
1	In the SnapManager for SQL Server main window, click the SnapManager Backup Wizard icon. **Result:** The SnapManager Backup Wizard starts and displays the Welcome screen.
Welcome	
2	Click Next. **Result:** The Databases to Back Up or Verify screen appears.
Databases to Back Up or Verify	
3	In the left panel, click to select the databases you want to verify, When you select a database, SnapManager automatically selects all other databases that reside on the same storage system volume. For information about deselecting any automatically selected databases, see the bullet "Which databases you want to back up" under "Job-specific parameters" on page 187.
4	Click Next. **Result:** The Back Up or Verify Databases and Transaction Logs screen appears.

Step	Action
Back Up or Verify Databases and Transaction Logs	
5	Specify the number of database backup Snapshot copies you want to verify: **a.** Select the "Verify Database in the" option. **b.** In the "most recent unverified backups" option, select the number of database backup Snapshot copies to verify. **c.** Click Next. **Result:** The Backup Management Group screen appears.
Backup Management Group	
6	Follow the instructions in the remainder of the Backup Wizard screens. The following screens enable you to specify the details of a database verification: ◆ Backup Management Group ◆ View or Change Verification Settings ◆ Advanced Backup Options ◆ Option to Perform SnapMirror Update After Operation **Note** This screen appears only if any SQL Server databases or SnapInfo directories reside on a SnapMirror source volume. ◆ Run a Command After the Operation For a list of information you need to provide, see "Information you need to specify for a database verification" on page 187.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">When to Run this Operation</td></tr>
<tr><td>7</td><td>The When to Run this Operation screen prompts you to choose whether you want the operation to be performed now or scheduled for a later time.<table><tr><th>If you want to...</th><th>Then...</th></tr><tr><td>Run the database verification now</td><td>Go to Step 8.</td></tr><tr><td>Schedule the database verification for later</td><td>Go to Step 11.</td></tr></table></td></tr>
<tr><td colspan="2">To run the database verification now:</td></tr>
<tr><td>8</td><td>If you want to run the database verification now, do the following:
a. Leave the Now option selected.
b. Click Next.
Result: The Completing the Backup Wizard screen appears and displays the database verification settings you have selected.</td></tr>
<tr><td>9</td><td>After you verify that all the settings in the window are correct, go to the Completing the Backup Wizard dialog box and click Finish.
Result: The Backup Wizard closes and the Backup Status window appears and displays a Backup Task List that will be used to show the progress of the database verification operation after you start it.</td></tr>
</table>

Step	Action
10	In the Backup Status window, click Start Now. **Result:** The database verification is performed. ◆ The Backup Status dialog box appears and displays a Backup Task List that will be used to show the progress of the backup operation after you start it. ◆ SnapManager Backup completes each task and checks it off on the list shown in the Backup Task List view. ◆ You can switch back and forth between the task check-off list and the progress report using the Switch buttons on either view. ◆ If the verification is successful, the Task window shows the check-off list with the tasks completed. ◆ If Notification is enabled, e-mail is sent and the event is posted to the Windows Application event log.
To schedule the database verification for later:	
11	If you want to schedule the database verification for later, do the following: **a.** Select the Later option. **b.** In the Job Name box, enter a name for this job. **c.** If you want this job name to overwrite a job of the same name (if it exists), select the Replace Job if Exists option. **d.** Click Next. For more information, see "Scheduling a backup job or a database verification job" on page 349. **Result:** The Select the Scheduling Service screen appears.

Step	Action
12	In the Select the Scheduling Service screen, do the following: **a.** Select either SQL Server Agent or Windows Scheduled Tasks. For more information, see "Scheduling a backup job or a database verification job" on page 349. **b.** Click Next. **Result:** The Completing the Backup Wizard screen appears and displays the backup settings you have selected.
13	After you verify that all the settings in the window are correct, go to the Completing the Backup Wizard dialog box and click Finish. **Result:** The selected scheduler displays one of two dialog boxes, as follows: ◆ If you chose to schedule the backup job using the SQL Server Agent, the Properties dialog box appears. ◆ If you chose to schedule the backup job using Scheduled Tasks, the Schedule Job dialog box appears.
14	To schedule the job and close the Backup Wizard, do the following: **a.** Specify the details of the job. **b.** Click OK.

Using backup management groups in backup and verification

When you create a full database backup, you have the option of assigning it to one of the backup management groups. The database Snapshot copy names and SnapInfo directory Snapshot copy names reflect the management group to which you assigned the backup. The purpose of backup management groups is to enable you to designate various levels of backup retention. See the following topics for more information:

- "How a backup is assigned a backup management group" on page 198
- "How backup management groups are used" on page 199
- "Changing the backup management group of an existing backup set" on page 200

How a backup is assigned a backup management group

When you create a backup, you can assign it to any one of the SnapManager backup management groups:

- Standard
- Daily
- Weekly

When you start or schedule a full database backup, the Backup Wizard and the Backup and Verification tab prepopulate the backup management group selection field with the Standard group.

For more information about starting or scheduling a full database backup, see the following topics:

- "Information you need to specify for a full database backup" on page 152
- "Full database backup using the Backup and Verification tab" on page 157
- "Full database backup using the Backup Wizard" on page 162

Note

The choice of backup management group—combined with the backup Snapshot copy naming convention selected *(unique* or *generic)*—affects the name assigned to the backup Snapshot copy. The name of each Snapshot copy created during a SnapManager backup operation includes information that identifies the Snapshot copy contents. This is described in "SnapManager Snapshot naming conventions" on page 142.

How backup management groups are used

The primary purpose of backup management group is to facilitate a database backup strategy. Backup management groups are used to determine which backups are targeted for automatic deletion of older backups, database verification for unverified backup Snapshot copies, and explicit deletion of backups.

Note

The backup management group neither depends on nor enforces how often backups are performed. Backup management groups are only a backup labeling convention that you can assign to a backup set.

Options for a full database backup: When you run or schedule a full database backup, you can specify how many of the most recent backups you want to retain. Only backups of the specified backup management group are deleted. The procedural details are included in "Creating a full database backup using SnapManager" on page 152.

Options for a database verification: When you run or schedule a database verification separate from the full database backup operation, you can limit the backups you want to verify by specifying a particular backup management group. The procedural details are included in "Performing database verification using SnapManager" on page 187.

Options for an explicit deletion of multiple backup Snapshot copies: When you explicitly delete multiple backups, you can specify that only backups belonging to a certain backup management group be deleted. The procedural details are included in "Deleting multiple backups" on page 203.

Backup example using backup management groups: Suppose your company wants to take regular backups between 7:30 a.m. and 7:30 p.m. You want to keep the last backup of the day and retain it for a few weeks, and you want to keep one backup per week for several months for archiving.

To achieve this using backup management groups, you could use the Standard backup management group for the backups during the day, and use a separate backup job to create one backup in the Daily management group at the end of the day. Then, once a week, you could use another job to create a backup in the Weekly backup management group.

You could then decide how many backups to retain independently for each backup management group. For example, you could keep 10 Standard backups, seven Daily backups (one week's worth), and four Weekly backups (one month's worth).

If your Daily or Weekly backup job failed for any reason, you could promote the most recent successful Standard backup to replace the Daily or Weekly backup by changing its backup management group.

Changing the backup management group of an existing backup set

Use the Change Backup Management Group dialog box to change the backup management group to which the selected backup set belongs.

To change the backup management group of an existing backup set, complete the following steps.

Note

You cannot change the backup management group of the most recent backup sets that were created using the Generic naming convention.

Step	Action
1	In the SnapManager main window, click the SnapManager Restore tab.
2	In the left panel, locate the backup set whose management group you want to change: ◆ Database Snapshots (Standard group) `sqlsnap__`*`sqlservername_date_time`* `sqlsnap__`*`sqlservername`*`__recent` ◆ Database Snapshots (Daily or Weekly group) `sqlsnap__`*`sqlservername_date_time__backupmgmtgroup`* `sqlsnap__`*`sqlservername`*`__recent__`*`backupmgmtgroup`* ◆ Snapshots of multiple databases on one or two LUNs `VDisk__`*`DriveLetter`* `VDisk__`*`FirstDriveLetter&SecondDriveLetter`*

Step	Action
3	Right-click on the backup set name to open a context menu, then select Change Management Group. **Result:** The Change Backup Management Group dialog box appears and displays the following information: ◆ The database name/LUN ◆ The Snapshot name ◆ The backups sharing this Snapshot copy ◆ The current management group
4	Carefully review the backups listed in the Backups Sharing This Snapshot list. **Note** The backup management group for all these backups is changed if you complete this operation. This is because they share a common Snapshot copy.
5	In the New Management Group list, select the backup management group you want to change to. **Note** When you change a backup's backup management group, you also change that backup's name, because the name includes the backup management group.
6	Click OK. **Result:** The backup management group for this backup and all backups listed in the All Backups Sharing This Snapshots list is changed. **Note** The report for the backup management group change is in the Miscellaneous report directory.

Explicitly deleting backup Snapshot copies using SnapManager

You can automatically delete older Snapshot copies by specifying the "Delete Oldest Backups In Excess Of" option in the SnapManager backup facility. This is the recommended method for managing the number of Snapshot copies stored. For more information, see "Automatic deletion of the oldest backups in a management group" on page 144.

You can also explicitly select the backup sets that you want to delete. For more information, see the following topics:

- "Understanding explicit deletion of backup Snapshot copies" on page 202
- "Deleting multiple backups" on page 203
- "Deleting a specific full database backup set" on page 207
- "Deleting a specific transaction log backup" on page 209
- "Busy Snapshot error prevents deletion of backup set" on page 210

Related topics:

- "Ways to manage the number of backup sets kept online" on page 144
- "Deleting restored Snapshot copies" on page 247

Understanding explicit deletion of backup Snapshot copies

SnapManager provides three ways for you to explicitly delete backup Snapshot copies.

If you want to delete...	Then use this method...
Multiple database Snapshot copies	In the Delete Databases dialog box, select the databases, the database component types, and the backup management group (Standard, Daily, Weekly, or All).
A specific full database backup set	In the SnapManager Restore tab, select the full database backup set you want to delete.
A specific transaction log backup	In the SnapManager Restore tab, select the transaction log backup you want to delete.

Each of the explicit deletion methods enables you to view detailed information about your selection before you proceed with the operation.

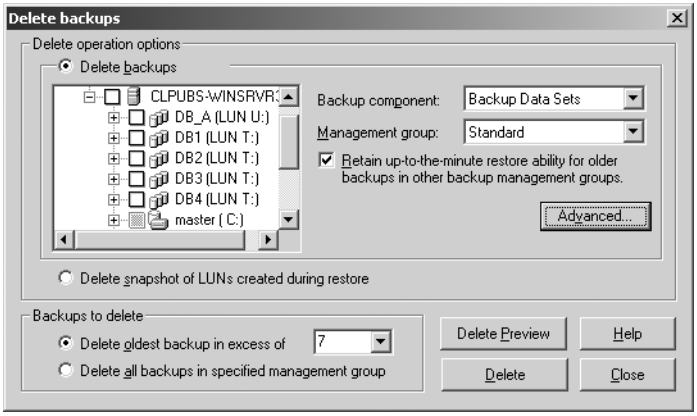
Deleting multiple backups

You can delete multiple backups for a specified group of databases by choosing which backup Snapshot copies you want to delete and whether you want to also delete LUN Snapshot copies created during the restore (if applicable).

Information you need to specify to delete multiple backups: An explicit deletion of multiple backups is specified using the following parameters.

- The backup Snapshot copies you want to delete
  - The databases for which you want to delete backups
  - The backup set components you want to delete: complete backup sets, transaction log backups only, or SnapInfo Snapshot copies only
  - The backup management group for which you want to delete backups: Standard, Daily, Weekly, or All
  - The number of backups to delete: all the backups in the specified management group or only the oldest backups, retaining only the number of backups specified
- Whether you want to also delete LUN Snapshot copies created during the restore

Procedure: To delete backups for a particular database set and backup management group, complete the following steps.

Step	Action
1	In the SnapManager main window, go to the menu bar and select Action > Delete Backup. **Result:** The Delete Backups dialog box appears. The Delete backups option is selected by default, and the Backups to Delete panel displays options pertaining to deleting backup Snapshot copies. 
2	In the database tree, select the databases for which you want to delete backup Snapshot copies.
3	The Backup Component option narrows the scope of the deletion by specifying the type of backup components to be deleted. This option is set to Backup Data Sets by default, but you can narrow this selection to transaction log backups only or to SnapInfo Snapshot copies only.
4	The Management Group option further narrows the scope of the deletion by specifying the backup management group to be deleted. This option is set to Standard by default, but you can change it to Daily, Weekly, or All.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>5</td><td>If you want to retain up-to-the-minute restore ability for older backups in other backup management groups, leave that option selected.
If you select this option, transaction logs will be deleted only from the specified backup management group; transaction logs will not be deleted from other backup management groups.
If you clear this option, transaction logs will be deleted from other backup management groups.</td></tr>
<tr><td>6</td><td>By default, only backups containing all of the selected databases will be deleted. You can override this behavior—for this particular backup deletion operation only—by using the Advanced button.<table><tr><th>If you want to...</th><th>Then...</th></tr><tr><td>Delete only backups containing all of the selected databases</td><td>Go to Step 9.</td></tr><tr><td>Delete backups containing any one or more of the selected databases</td><td>Go to Step 7.</td></tr></table></td></tr>
<tr><td>7</td><td>Click Advanced.

Result: The Advanced Options dialog box appears.</td></tr>
<tr><td>8</td><td>Click OK to apply your change and close the dialog box.

Note
For this backup deletion operation only, multiple backup deletions will delete backups containing any one or more of the selected databases.</td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="4">9</td><td colspan="2">Specify which Snapshot copies you want to delete.</td></tr>
<tr><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Delete only the oldest restore Snapshot copies</td><td>1. Select the Delete Oldest Snapshot copies In Excess Of option.
2. Specify how many of the newest restore Snapshot copies you want to preserve.</td></tr>
<tr><td>Delete all the restore Snapshot copies</td><td>Select the Delete All Snapshots Created During Restore option.</td></tr>
<tr><td rowspan="4">10</td><td colspan="2">You can delete the selected restore Snapshot copies immediately, or you can first view the list of restore Snapshot copies that are targeted for deletion.</td></tr>
<tr><td>If you want to...</td><td>Then...</td></tr>
<tr><td>View the list of restore Snapshot copies that would be deleted</td><td>Go to Step 11.</td></tr>
<tr><td>Delete the restore Snapshot copies</td><td>Click Delete.
Result: The restore Snapshot copies identified by your selections are deleted. When the deletion is complete, a status message is displayed. You have completed this procedure.</td></tr>
<tr><td>11</td><td colspan="2">Click Delete Preview.
Result: The Delete Backup Data Set Preview dialog box appears. After a moment, the dialog box displays a count and list of the restore Snapshot copies identified for deletion.
If you want to view a report, click Show Report.</td></tr>
</table>

Step	Action	
12	Based on the list displayed in the Delete Backup Data Set Preview dialog box, you can cancel the delete operation or proceed with the delete operation.	
	If you want to...	**Then...**
	Cancel the operation	**1.** Click Close to close the Delete Backup Data Set Preview dialog box. **2.** Click Close to close the Delete Backups dialog box.
	Delete the restore Snapshot copies listed in the preview	**1.** Click Close to close the Delete Backup Data Set Preview dialog box. **2.** Click Delete on the Delete Backups dialog box.

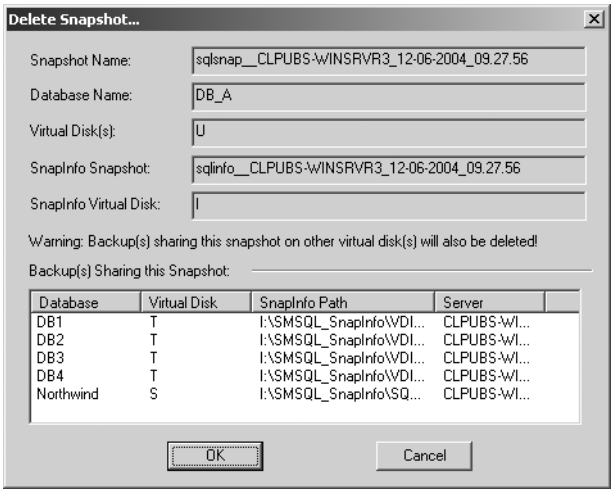
Deleting a specific full database backup set

You can delete a specific full database backup set selected in the SnapManager Restore tab.

Information needed to delete a full database backup set: To select a full database backup for explicit deletion, select the backup set in the SnapManager Restore tab.

Procedure: To delete a specific full database backup, complete the following steps.

Step	Action
1	In the SnapManager main window, click the SnapManager Restore tab.
2	In the database tree in the left panel, select the full database backup Snapshot copy you want to delete. **Result:** The right panel displays detailed information about the selected full database backup set.

Step	Action
3	In the SnapManager main window, go to the menu bar and select Action > Delete Full Database Backup. **Result:** The Delete Snapshot dialog box appears and displays information about the selected Snapshot copy, as well as a list of all database backup sets that share the same Snapshot copy. 
4	Click OK to delete the selected Snapshot copy. **Result:** A message box asks you to confirm that you want to proceed with deleting the Snapshot copy.
5	Click OK to confirm that you do in fact want to delete the selected Snapshot copy. **Result:** The message box closes, the Delete Snapshot dialog box closes, and SnapManager deletes the selected Snapshot copy.

Deleting a specific transaction log backup

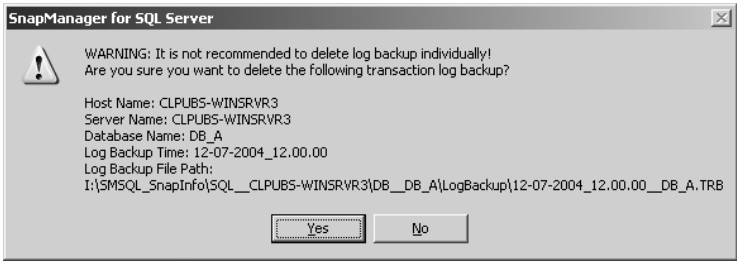
You can delete a specific transaction log backup selected in the SnapManager Restore tab.

Note

NetApp recommends that you do not delete a transaction log backup, because doing so will prevent you from performing an up-to-the-minute restore of the database using that backup set.

Information needed to delete a transaction log backup: To select a transaction log backup for explicit deletion, select the log backup in the SnapManager Restore tab.

Procedure: To delete a specific full database backup, complete the following steps.

Step	Action
1	In the SnapManager main window, click the SnapManager Restore tab.
2	In the database tree in the left panel, select the full database backup Snapshot copy associated with the transaction log backup you want to delete. **Result:** The right panel displays detailed information about the selected full database backup set.
3	In the right panel, select the transaction log backup you want to delete.
4	In the SnapManager main window, go to the menu bar and select Action > Delete Transaction Log Backup. **Result:** The following message box appears. 

Step	Action
5	Click Yes to close the message box and proceed with deleting the selected Snapshot copy. **Result:** Another message box asks you to confirm that you want to proceed with deleting the Snapshot copy.
6	Click OK to confirm that you do in fact want to delete the selected Snapshot copy. **Result:** The message box closes, the Delete Snapshot dialog box closes, and SnapManager deletes the selected Snapshot copy.

Busy Snapshot error prevents deletion of backup set

If you attempt to delete a backup and SnapManager displays an error message stating that the Snapshot copy is busy and cannot be deleted, you have a backup that was taken of a LUN backed by another Snapshot copy.

Definition of a busy Snapshot copy: A Snapshot copy is in a busy state if there are any LUNs backed by data in that Snapshot copy. The Snapshot copy contains data that is used by the LUN. These LUNs can exist either in the active file system or in some other Snapshot copy. For more information about how a Snapshot copy becomes busy, see the *Block Access Management Guide* for your version of Data ONTAP.

If you attempt to delete a busy Snapshot copy: If you initiate a backup when a LUN backed by Snapshot copy exists, the result is a Snapshot copy that cannot be deleted; if you do attempt to delete the Snapshot copy, the following events occur:

- SnapManager displays a “busy Snapshot” error message.
- SnapDrive logs event 249 in the Windows application event log.

To check whether you have a busy Snapshot copy: There are two ways to determine whether you have a busy Snapshot copy:

- View your Snapshot copies in FilerView.
- Use the following filer command to list the busy Snapshot copies.

  `snap list usage` *VolumeName BusySnapshotName*

  The full description of the preceding command syntax is described in the *Command Reference* for your version of Data ONTAP.

To delete a busy Snapshot copy: Delete the more recently taken backup; then delete the older backup. For more information about deleting a busy Snapshot copy, see the *Block Access Management Guide* for your version of Data ONTAP.

To avoid this situation in the future: Avoid performing SnapManager backups while you have any LUNs backed by Snapshot copies.

- During a *database verification*, a LUN in Snapshot copy is mounted and the DBCC utility is run against the database. For this reason, it is important to carefully plan your SnapManager backup and verification schedules. See "Recommendations for scheduling backups" on page 148.
- While *archiving from a LUN backed by a Snapshot copy*, avoid performing a SnapManager backup.

Restoring Databases Using SnapManager 10

This section describes how to recover your SQL Server databases from SnapManager backup sets. The following topics are covered:

- "SQL Server 2000 and 2005 recovery models" on page 214
- "Understanding SnapManager Restore" on page 216
- "Types of SnapManager restore operations" on page 222
- "Choosing the type of restore operation to perform" on page 224
- "Performing a restore operation" on page 225
- "Managing databases mounted at or restored to alternate locations" on page 242
- "Deleting restored Snapshot copies" on page 247

Note

If you are running SQL Server 2000, SnapManager does not allow you to restore a backup set created using SQL Server 2005.

Attention

You must run SnapManager from the system console, not from a Terminal Services client. Do not use Terminal Services for any type of SnapManager administration, because you might miss critical information that is displayed only in pop-up boxes at the system console.

SQL Server 2000 and 2005 recovery models

SnapManager 2.1 for SQL Server supports all three types of SQL Server recovery models. See the following topics for more information:

- "Recovery models supported by SnapManager" on page 214
- "Implications for SnapManager operations" on page 215

Recovery models supported by SnapManager

SnapManager for SQL Server supports all three types of SQL Server recovery models:

- SIMPLE
- FULL
- BULK_LOGGED

The SQL Server database administrator can assign each database a different recovery model, but specific recovery models are assigned to each database type by default.

SQL Server system database type	Default recovery model
master	SIMPLE
tempdb	SIMPLE
model	FULL
msdb	SIMPLE
distribution	FULL

The recovery model defines the fault tolerance level of your SQL Server environment. For more information about SQL Server recovery models, see the following resources:

- The description of "recovery model" on page 10
- Your Microsoft SQL Server documentation

Implications for SnapManager operations

The recovery model of an SQL Server database affects SnapManager operations, as described in the following paragraphs.

SIMPLE recovery model: When the SIMPLE recovery model is used, transaction logs cannot be backed up.

FULL recovery model: When the FULL recovery model is used, you can restore a database to its state at the point of failure. This entails the following sequence:

1. Back up the current active transaction log (if possible).
2. Restore the most recent database backup without recovery.
3. Restore each transaction log backup since the last restored backup.
4. Restore the transaction log backup of the currently active transaction log.

Note

If you want to do a fully recovery of the master database, deselect the Run transaction log backup after full database backup option below the Backup Management group for successful backup.

BULK_LOGGED recovery model: When the BULK_LOGGED recovery model is used, you need to manually redo bulk operations under the following circumstances:

- Before you can perform a full database backup
- Before you can perform a transaction log backup
- If the database fails

Understanding SnapManager Restore

SnapManager Restore restores the SQL Server databases you select. For more information, see the following topics:

- "Sources for a restore operation" on page 216
- "Destinations for a restore operation" on page 217
- "Databases restored to or mounted at temporary, alternate locations" on page 217
- "How SnapManager Restore works" on page 219
- "Importance of verifying databases to be restored" on page 221

Related topics:

- "Types of SnapManager restore operations" on page 222
- "Choosing the type of restore operation to perform" on page 224

Sources for a restore operation

SnapManager Restore enables you to restore databases from a SnapManager backup set.

Restore from a SnapManager backup set: You can restore databases from SnapManager backup sets created for the same SQL Server instance or created for a different instance. The LUNs containing the selected SQL Server's databases are restored from the backup.

If SQL Server system databases fail, they can be restored from stream-based SnapManager backup sets of those databases. For more information, see "Restoring system databases from SnapManager backup sets" on page 337.

Restore from an archive of SnapManager backup sets: You can also use SnapManager Restore to restore databases from offline archives of Snapshot copies containing SnapManager backup sets.

- For information about creating offline archives of Snapshot copies containing SnapManager backup sets, see Chapter 13, "Archiving SnapManager Backups," on page 267.
- For information about recovering SQL Server databases from archives, see "Recovering SQL Server databases using archives" on page 309.

Restore an online database as a new database: You can use SnapManager Restore to restore an online database as a new database on the *same SQL Server instance*. However, you *cannot* restore an online database as a new database on a *different SQL Server instance*.

Destinations for a restore operation

You can restore databases to various types of destinations.

Restore to the original database location: By default, SnapManager restores to the same *database names* and on the same *SQL Server instance*.

Restore to an alternate location: Instead, you can use any combination of the following options to restore databases to alternate locations:

- Restore to different database names
- Restore to a different SQL Server instance on the same or different SQL Server computer

Restore to a temporary, alternate location: The database is mounted at a temporary, alternate location using a writable Snapshot copy, and then the transaction logs are applied. Because the logs are applied, the data is current.

Mount at a temporary, alternate location without restoring: The database is mounted at a temporary, alternate location, but the transaction logs are not applied. Because the data is not current, you should use this function to view only the layout of the data.

Databases restored to or mounted at temporary, alternate locations

The following information applies to databases that have been *restored to* or only *mounted at* temporary, alternate locations using writable Snapshot copy.

Backup set label: A database restored to or mounted at a temporary, alternate location is listed in the Backup and Verification tab with the following label:

`SnapLUN`

Backup set name: If the database name already exists on the server, or if the backup set consists of more than one database, the database is listed using the following naming convention:

databasename`__Mount`

Avoiding a busy Snapshot condition: Explicitly detach the temporary database and dismount the Snapshot copy LUNs as soon as you finish viewing the data or data layout, otherwise you might encounter a *busy Snapshot* condition when you attempt to delete the Snapshot copy. There are two ways to do this:

- If the database was *restored* with a post-restore state of *Read-Only* or is in the *loading* state and you cannot bring the database into read-write mode, use SQL Server Enterprise Manager or Management Studio and SnapDrive.
- Otherwise, use a SnapManager function that is available from the Backup and Verification tab.

For a detailed description, see “Detaching a database and dismounting Snapshot LUNs” on page 245.

How SnapManager Restore works

The following list summarizes the steps that SnapManager performs when restoring databases from a backup set:

1. SnapManager Restore checks the SnapManager license.
2. If the "Create transaction log backup before restore" option is selected, the transaction log is backed up before the restore is performed.

   Note

   If you are restoring to a temporary, alternate location using a writable Snapshot copy, a transaction log backup will *not* be created before the restore, even if this option is selected. If you want to create a transaction log backup, do so as a separate operation before you restore to the alternate location.

   For reasons to deselect this option, see "Understanding the restore options" on page 382.

3. All of the databases that reside on the LUN in a selected backup set are restored to the active file system. The restore method used by SnapManager depends on (1) the method that was used to create the backup set and (2) the specific subset of databases you choose to restore from the backup set.

   SnapManager uses the *stream-based* restore method if you are restoring from a stream-based backup set. With this method, each of the databases is restored individually. Depending on the composition of the backup set, a stream-based restore could require additional time and free space on the storage system as compared to an online Snapshot copy restore.

   SnapManager uses the *online Snapshot* copy restore method if you are restoring from a backup set that contains all of the databases that reside on the same LUN. Starting with Data ONTAP 7.1, online Snapshot copy restore speed increased significantly, due to the replacement of the single-file SnapRestore method (SFSR) with the LUN clone split restore method.

   SnapManager uses the *copy-based* restore method if any of the following conditions are true:

   - The backup set contains only a subset of the databases that reside on the same Data ONTAP LUN (not recommended).
   - You select only a subset of the databases contained in the backup set.
   - A new database was added to the same LUN after the backup was created.
   - The backup was split when the number of databases sharing one or two LUNs exceeded the maximum concurrent database setting. For more information, see "Using the Number of Databases that Can Be Backed Up Concurrently screen" on page 417.

4. If the database has transaction log backups, SnapManager Restore can apply the transaction log backups (if necessary).
5. Depending on the database restore option selected, SnapManager Restore performs a *point-in-time* restore or an *up-to-the-minute* restore.

Restore Snapshot copies: Every time you perform a restore operation using SnapManager, SnapManager first creates a Snapshot copy on each storage system volume that contains files for the databases you will be restoring. That way, in the unlikely event that a catastrophic failure occurs during a restore, you have recent Snapshot copies of the LUNs that can be used to re-create those databases as they existed prior to the start of the failed restore operation.

Each restore Snapshot copy is named using the following naming convention:

`rstrsnap__`*SqlServerName_date_time*

The Snapshot copy name contains the name of the SQL Server instance to which the backup was restored (indicated by the variable *SqlServerName*) and the Snapshot copy creation date and time (indicated by the variable *date_time*).

After you verify that a restore was completed successfully and you are satisfied with the results, you can delete the restore Snapshot copy. For details, see "Deleting restored Snapshot copies" on page 247.

SQL Server cluster group state during a restore: If you are using SnapDrive 4.1, SnapManager 2.1 can restore databases in a Windows cluster without taking the SQL Server cluster group offline while restoring a LUN. This requires that you have set a registry variable on the Windows host so that SnapDrive can restore the MSCS LUN without taking the resource offline. This requirement is described in the procedure "Installing SnapManager and creating a new Windows cluster" on page 49.

This feature was introduced in SnapManager 2.0 and is available only if you are using SnapDrive 3.2 and higher versions.

Cluster failure during a restore operation: If a cluster failure (a cluster group move operation) occurs during the restore operation—for example, if the node that owns the resources goes down—you must reconnect to the SQL Server instance, restart the SnapManager user interface, and then restart the restore operation.

Transaction log restore operations: A SnapManager transaction log restore uses the SQL recovery process to play forward transactions from the log backup into the restored database.

Importance of verifying databases to be restored

The database verification process protects you from restoring a backup that contains any physical-level corruption. Physical-level database corruption can occur silently in SQL Server databases. The only way to know whether a particular database backup incurred physical-level corruption is to run database verification on that backup.

Before allowing a restore operation to proceed, SnapManager enables you to check that the selected backup set was verified through the use of DBCC CHECKDB.

Backup verification status: SnapManager Restore shows you a list of the backups that have been taken. For each backup, the date and time of the backup is displayed, as well as an icon that indicates the backup verification status.

Icon description	Backup verification status
Green-circled black check mark	The databases in this backup have been verified.
Yellow-circled black question mark	The databases in this backup have not been verified.

Note

If you select a database on which a consistency check has not been run successfully, SnapManager prompts (but does not require) you to run DBCC before performing a restore. Running database consistency checking as part of recovery increases the time the recovery takes.

Types of SnapManager restore operations

You can use SnapManager to perform either of the following types of restore operations:

- "Up-to-the-minute restore operation" on page 222
- "Point-in-time restore operation" on page 223

Related topics:

- "Understanding SnapManager Restore" on page 216
- "Choosing the type of restore operation to perform" on page 224

Up-to-the-minute restore operation

In an up-to-the-minute restore, databases are recovered up to the point of failure. SnapManager accomplishes this by performing the following sequence:

1. The last active transaction log is automatically backed up.
2. The databases are restored from the full database Snapshot copies you select.
3. All the transaction logs that were not committed to the databases, from the time the backup set was created up to the most current time, are played forward and applied to the databases (if selected). This includes transaction logs from the *backup sets*, as well as the existing transaction logs in the *transaction log directory.*

An up-to-the-minute restore requires a *contiguous set of transaction logs*. The up-to-the-minute restore type is selected by default. For more information, see "Choosing the type of restore operation to perform" on page 224.

Note

Because SnapManager cannot restore transaction logs from *log-shipping* backup files, you might not be able to restore the database using an up-to-the-minute restore. For this reason, NetApp recommends that you use SnapManager alone to back up your SQL Server database transaction log files.

Example: Suppose you run SnapManager Backup every day at noon, and on Wednesday at 4 p.m. you need to restore from a backup. For some reason, the backup set from Wednesday lunchtime failed verification, so you decide to restore from the Tuesday lunchtime backup. After that backup is restored, all the transaction logs are played forward and applied to the restored databases, starting with those that were not committed when you created Tuesday's backup set and continuing through the latest transaction log written on Wednesday at 4 p.m.

Point-in-time restore operation

In a point-in-time restore, databases are restored only to a point-in-time from the past. If you choose not to apply any transaction log backup sets, the transaction logs beyond the point-in-time that exist in the transaction log directory belonging to the restored database are removed.

You can also choose to apply a contiguous set of transaction log backup sets. If you apply transaction log backup sets, you can also specify a particular date and time at which you want to stop the application of backed up transactions. To do this, you specify a date and time within the available range and SnapManager will roll back any transactions that were not committed prior to that point in time. You can use this method to restore databases back to a point in time before a corruption occurred. For more information about choosing the restore type, see "Choosing the type of restore operation to perform" on page 224.

Example: Suppose you take full database backups once at midnight and a transaction log backup every hour. The database crashes at 9:45 a.m, but you still back up the transaction logs of the failed database. You can choose from among three point-in-time restore scenarios:

1. Restore the full database backup taken at midnight and accept the loss of the database changes made afterward.
2. Restore the full database backup and apply all the transaction log backups until 9:45 a.m.
3. Restore the full database backup and apply transaction log backup sets. Specifying the time you want the transactions to restore from the last set of transaction log backups.

   In this case, you would calculate the date and time where a certain error was reported. Any transactions that were not committed prior to the date and time specified in the `Restore` command are rolled back.

Note

You can use this method to restore the database up to a point where you think corruption occurred, or to recover from an accidental database or table deletion.

Choosing the type of restore operation to perform

Use following general guidelines to help you decide whether to use a point-in-time restore or an up-to-the-minute restore:

- "Capabilities" on page 224
- "Requirements and limitations" on page 224
- "Point-in-time restore from the most recent backup" on page 224

Related topics:

- "Understanding SnapManager Restore" on page 216
- "Types of SnapManager restore operations" on page 222

Capabilities

Often you can choose a restore type based on the particular restore capabilities needed.

- If you want to roll forward all the transactions up to the most current time, use an up-to-the-minute restore.
- If you want to recover the databases as they were at a particular point in time, for example, at the point when the most recent backup was created, use a point-in-time restore.

Requirements and limitations

Before choosing a restore type, be aware of the requirements and limitations of each.

- For to an up-to-the-minute restore to succeed, a contiguous set of all required transaction log backups must be in the SnapInfo folder.
- After a point-in-time restore of a backup that is not the most recent one, all existing backups become point-in-time restorable only. Backups created after the restore will be available for up-to-the-minute restores.

Point-in-time restore from the most recent backup

Use caution when selecting a point-in-time restore from the most recent backup. After a point-in-time restore of the most recent backup, you can still perform an up-to-the-minute restore from any backup, including the most recent backup. However, any transactions that occurred between when you created the last backup and before you performed the point-in-time restore will not be restored. The reason for this is that when you perform the point-in-time restore, the transaction logs containing transactions recorded after the selected backup are removed.

Performing a restore operation

You can restore SQL Server databases from a SnapManager backup in two ways: using the SnapManager Restore tab or using the SnapManager Restore Wizard. See the following topics for more information:

- "Preparing to perform a restore operation" on page 225
- "Information you need to perform a restore" on page 226
- "Restoring using the SnapManager Restore tab" on page 229
- "Restoring using the SnapManager Restore Wizard" on page 236

Related topics:

- "Managing databases mounted at or restored to alternate locations" on page 242
- "Deleting restored Snapshot copies" on page 247

Preparing to perform a restore operation

Preparing to restore operation from a SnapManager backup set:

Before you restore from a SnapManager backup set, review the following checklist:

- The SQL Server must be online and running before a SnapManager Restore can take place. This applies to both user database restore operations and system database restore operations.
- Be sure that the target databases are detached or in a suspect state.

Note

You can perform a restore of the SQL Server databases with the databases online, but this requires that the online restore option be enabled. To view or change this option and other options pertaining to the SnapManager restore operation, go to the SnapManager main window and select Options > Restore Settings.

Follow these guidelines when restoring a SnapManager backup set:

- Although not necessary, it is recommended that you always restore from the most recent Snapshot copy, `sqlsnap__`*SqlServerName*`_recent`, where *SqlServerName* is the host name of the SQL Server.
- If you rename a database, make sure that you back it up as soon as possible.

Note

If you use SnapManager to restore a backup that is not the most recent one, that backup's sequential backup sets are still available for future restore.

Preparing to restore an online database as a new database: You must detach the online database before you begin restoring it as a new database on the *same* SQL Server instance.

Information you need to perform a restore

A SnapManager backup set restore operation is specified using a combination of parameters.

Job-specific parameters: Each time you restore a SnapManager backup set, you must specify the following information in either the SnapManager Restore tab or in the Restore Wizard:

- Backup set from which the databases are to be restored

  You can select an *unverified* backup set, but SnapManager will ask you to confirm your selection. NetApp strongly recommends that you restore only from verified backup sets.

  If for some reason you do not have a verified backup set available when you need to perform a restore and you do not want to wait for a verification to be completed before you perform the restore, you might find it necessary to restore directly from an unverified backup set.

  Note

  If you must restore from an unverified backup set, NetApp recommends that you perform an *up-to-the-minute* restore operation. This way, if you discover later that the backup set was corrupted, you can restore the database from a different backup set.

  If an SQL Server 2005 database has the full-text search option enabled, the *full-text search catalogs* are visible when you click the "+" next to the database name. The full text catalogs can be migrated, backed up, and restored along with the other files or filegroups of the database.

- Databases to be restored from the backup set:
  - All databases in the backup set (the default setting)
  - A subset of the databases in the backup set

  To choose only a subset of the databases in the selected backup set, highlight any database in the right pane and then select the Unselect All Databases and Logs item from the context menu. This deselects all databases in the backup set. You can then choose the individual databases that you want to restore.

- Database target:
  - Original database (the default setting)
  - A database of a different name

- SQL Server instance to which the backup set is to be restored:
  - Original SQL Server (the default setting)
  - A different SQL Server (only on the same host)
- Restore type:
  - Up-to-the-minute (the default setting)
  - Point-in-time

  For more information, see "Types of SnapManager restore operations" on page 222 and "Choosing the type of restore operation to perform" on page 224.

  Note

  You cannot restore multiple databases with different restore options in a single restore operation.

- Restore location:
  - Original database (the default setting)
  - Alternate location using a writable Snapshot copy
- The transaction log backup sets to be restored

  Note

  If you are restoring a *log-shipped* database, do not restore the transaction logs. Restoring the transaction logs to a log-shipped database causes the SnapManager operation to fail.

- The state to which the databases are to be set after the restore operation finishes
  - For a single-database restore operation, this is configured in the Restore Options dialog box, described in "Specifying the post-restore state of a single database" on page 437.
  - For a multiple-database restore operation, this is configured in the Multiple Database Restore Options dialog box, described in "Specifying the post-restore state of multiple databases" on page 441.

Verification settings: The following list summarizes the settings that pertain to database restore operations:

- Which SQL Server is used to perform database verification

  This is configured using the SQL Server tab of the Verification Settings dialog box. See "Selecting the database verification server" on page 372.
- Which DBCC options are used to verify database backup Snapshot copies

This is configured using the DBCC Options tab of the Verification Settings dialog box. See "Selecting DBCC options" on page 374.

The Verification Settings dialog box can be accessed from the SnapManager Restore tab, as well as from the Restore Wizard.

- If you are using the SnapManager Restore tab, the Verification Settings dialog box opens automatically after you click Verify Now.
- From within the Restore Wizard, you can open the Verification Settings dialog box by clicking Database Verification Settings in the "View or Change Database Verification Options" screen.

Restore settings: The following restore settings determine how SnapManager is to restore database backup Snapshot copies:

- Recover databases without restoring at the end of the restore, if needed
- Restore databases even if existing databases are online
- Retain SQL database replication settings
- Create transaction log backup before restore

Note

If you are restoring a *log-shipped* database, disable the option to create a transaction log backup before the restore.

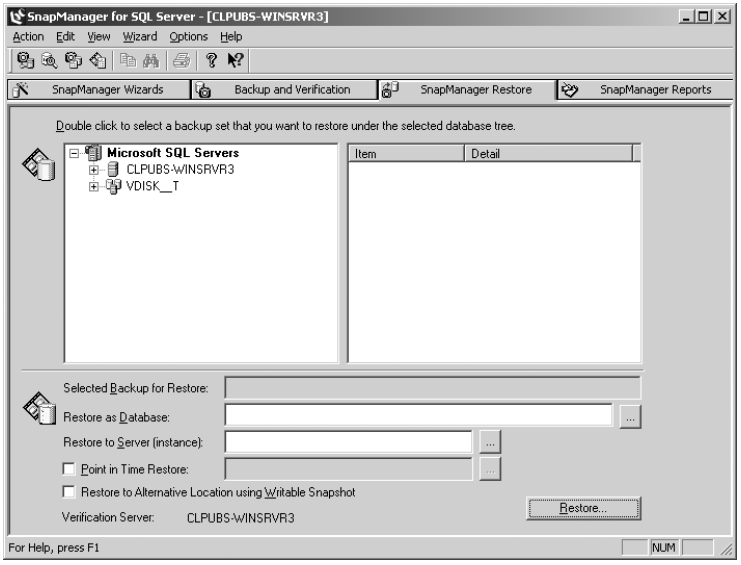
- Abort database restore if transaction log backup before restore fails

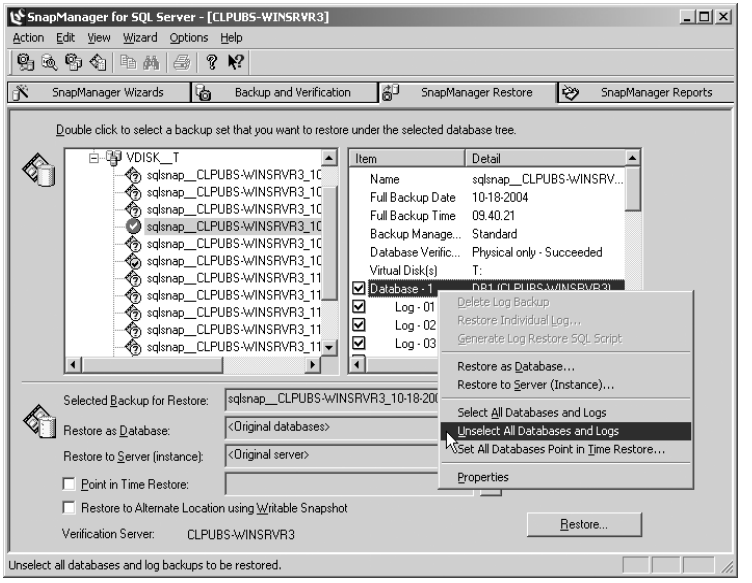
These settings are configured using the Restore Settings dialog box, described in "Configuring the profile of a restore operation" on page 383.

Restoring using the SnapManager Restore tab

To restore an SQL Server database from a backup set using the SnapManager Restore tab, complete the following steps.

Step	Action
1	Review the list in "Information you need to perform a restore" on page 226.
2	Make sure that all Windows Explorer windows are closed on the SQL Server computer that is running SnapManager.
3	Disable any SnapManager operations that are scheduled to run against the SQL Server data you are restoring, including any jobs scheduled on remote management or remote verification servers.

Step	Action
4	In the SnapManager main window, click the SnapManager Restore tab. **Result:** The tab enables you to navigate the backup sets from which you can select databases to be restored. 
Specify the source.	
5	If you want to restore databases to this SQL Server using SnapManager backup sets that were created for other SQL Servers, follow the procedure described in "Restoring from other SQL Server backups using the SnapManager Restore tab" on page 326.
6	In the left panel, double-click to select the backup set from which you want to restore. **Result:** The Selected Backup for Restore field displays the backup set you selected to be restored.

Step	Action
7	If you selected a multiple-database backup set, you can follow these steps to select only a subset of the databases: **a.** In the right panel, highlight any database. **b.** Right-click to open the context menu, then select Unselect All Databases and Logs to deselect all of the databases in the backup set.  **c.** To select an individual database in the backup set, click to select that database in the right panel. **Result:** Only the selected databases will be restored from the backup sets you specified.

Step	Action
Specify the destination.	
8	If you want to restore to an SQL Server other than the original, follow these steps: **a.** Click "..." to the right of the "Restore to Server (instance)" box. The "Select SQL Server to Restore..." dialog box appears. **b.** Select the SQL Server to which you want to restore the database. **c.** Click OK to apply your change and close the dialog box.
9	If you want to restore to a database other than the original database, follow these steps: **a.** If you selected a multiple-database backup set, click in the right panel to select a single database that you want to restore to a database other than the original. **b.** Click "..." to the right of the Restore as Database box. The "Individual Database Restore As..." dialog box appears. **c.** In the Restore as Database box, enter the database name to which you want the backup restored. This database name must not already exist on the SQL Server instance to which you will be restoring the database. **d.** Click OK to apply your change and close the dialog box.

Step	Action
10	If you want to restore the databases to a temporary, alternate location using a writable Snapshot copy, select the "Restore to Alternate Location using Writable SnapShot" option. For more information, see "Destinations for a restore operation" on page 217. **Note** As soon as you have finished viewing the data, detach the database and dismount the Snapshot copy LUNs as described in "Detaching a database and dismounting Snapshot LUNs" on page 245.
Specify the restore type.	
11	If you are restoring a *single database* and you want to perform a *point-in-time* restore, do the following: **a.** Select the Point in Time Restore option. The Point in Time Restore dialog box appears. **b.** In the Point in Time Restore dialog box, specify the date and time after which transaction logs will not be applied to the restored database. **c.** Click OK to apply your change and close the dialog box. If you want to view or change the date and time for the point-in-time restore, click "..." to the right of the "Point in Time Restore" box. **Note** A point-in-time restore will halt the restoration of transaction log entries that were recorded after the specified date and time.

Step	Action
12	If you are restoring *multiple databases* and you want to perform a *point-in-time restore* of all of the databases to the point in time when the backup was created, do the following: **a.** Select the Point in Time Restore option. A message box appears and prompts you to confirm that you want to restore all the selected databases to the point in time when the backup was created. **b.** In the message box, click Yes.
13	If you are restoring *multiple databases* and you want to perform a *point-in-time* restore of all of the databases to the point in time when the backup was created followed by a *transaction log* restore, do the following: **a.** Select the Point in Time Restore option. A message box appears and prompts you to confirm that you want to restore all the selected databases to the point in time when the backup was created. **b.** In the message box, click No. Another message box appears and prompts you to confirm that you want to restore all the selected databases to the point in time when the backup was created and then also restore from a transaction log backup. **c.** In the message box, click Yes. The Point in Time Restore dialog box appears. **d.** In the Point in Time Restore dialog box, specify the date and time after which transaction logs are not to be applied to the restored databases. **e.** Click OK to apply your change and close the dialog box. If you want to view or change the date and time for the point-in-time restore, click "..." to the right of the Point in Time Restore box.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>14</td><td>Click Restore to open a dialog box that enables you to finish specifying the restore operation.

Note
A different the dialog box will appear, depending on whether you are restoring a single database or multiple databases.

<table><tr><th>If...</th><th>Then...</th></tr><tr><td>You are restoring a single database</td><td>The Restore Options dialog box appears.

For detailed instructions, see “Specifying single-database recovery state from the SnapManager Restore tab” on page 437.</td></tr><tr><td>You are restoring multiple databases</td><td>The Multiple Databases Restore Option dialog box appears.

For detailed instructions, see “Specifying multiple-database recovery state from the SnapManager Restore tab” on page 441.</td></tr><tr><td colspan="2">Result: The Restore Status dialog box appears and displays the Restore Task List, which will be used to show the progress of the restore operation after you start it.</td></tr></table></td></tr>
</table>

Step	Action
15	To start the restore operation, click Start Now. **Result:** SnapManager begins to restore your databases from the backup you selected. SnapManager Restore completes each task and checks it off the list shown in the Restore Task List view. You can switch back and forth between the task check-off list and the progress report using the Switch buttons on either window. If the restore is successful, the Task window shows the check-off list with the tasks completed, and a dialog box reports that the restore was successful. **Note** If Notification is enabled, e-mail is sent and the event is posted to the Windows event log.
16	After all the restore tasks are finished, click OK. **Result:** Your restore is complete and your SQL Server computer comes back online.
17	After the restore is finished, you can optionally perform a SnapManager backup and verification to verify that your restored database is free of physical-level corruption.

Restoring using the SnapManager Restore Wizard

To restore an SQL Server database from a backup set using the SnapManager Restore Wizard, complete the following steps.

Step	Action
1	Review the list in “Information you need to perform a restore” on page 226.
2	Make sure that all Windows Explorer windows are closed on the SQL Server computer running SnapManager.
3	Disable any SnapManager operations that are scheduled to run against the SQL Server data you are restoring, including any jobs scheduled on remote management or remote verification servers.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>To launch the SnapManager Restore Wizard, go to the SnapManager menu bar and select Wizard > SnapManager Restore.

Result: The SnapManager Restore Wizard appears and displays the Welcome screen.</td></tr>
<tr><td colspan="2">Welcome</td></tr>
<tr><td>5</td><td>Click Next.

Result: The Which SQL Server Created the Backups screen appears.</td></tr>
<tr><td colspan="2">Which SQL Server Created the Backups</td></tr>
<tr><td>6</td><td>By default, SnapManager restores from backups that were created on this SQL Server.
<table>
<tr><th>If...</th><th>Then...</th></tr>
<tr><td>Restore from backups that were created on this SQL Server</td><td>Do the following:

a. Leave the “Restore SnapManager backups that were created on a different SQL Server” option unselected.

b. Click Next.

c. Go to Step 8.</td></tr>
<tr><td>Restore from backups that were created on a different SQL Server</td><td>Follow the procedure described in “Restoring from other SQL Server backups using the SnapManager Restore Wizard” on page 331.</td></tr>
</table></td></tr>
<tr><td colspan="2">SQL Server Where the Backups Were Created</td></tr>
<tr><td>7</td><td>The “SQL Server Where the Backups Were Create” screen appears only if you selected the “Restore from Another SQL Server Backup” option. Follow the procedure described in “Restoring databases from other SQL Server backups” on page 326.</td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td colspan="3">Single-database backup set or multiple-database backup set?</td></tr>
<tr><td>8</td><td colspan="2">The sequence of Restore Wizard screens varies, depending on whether you choose to restore from a single-database backup set or from a multiple-database backup set.</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You are restoring from a single-database backup set</td><td>Go to Step 9.</td></tr>
<tr><td></td><td>You are restoring from a multiple-database backup set</td><td>Go to Step 11.</td></tr>
</table>

Step	Action
Restoring from a single-database backup set	
9	Follow the instructions in the Restore Wizard screens: ◆ Backup to Restore In the left panel, expand the list, then double-click to select the single-database backup set from which you want to restore. In the right panel, you can deselect or reselect transaction log backups. ◆ Restore Transaction Log Backup This screen appears only if you selected transaction log backups. If you do not want to restore transaction log backups, then click Back, and deselect the transaction log backups. ◆ Type of Restore Select either Up-to-the-Minute restore or Point-in-Time restore. ◆ Database State after the Restore For detailed instructions, see "Specifying single-database recovery state from within the Restore Wizard" on page 438. ◆ Restored Database Name ◆ SQL Server Where the Backup Will Be Restored If you want to restore to an alternate location using a writable Snapshot copy, select the "Restore to Alternate Location using Writable SnapShot" option. For more information, see "Destinations for a restore operation" on page 217. As soon as you have finished viewing the data, detach the database and dismount the Snapshot copy LUNs. See "Detaching a database and dismounting Snapshot LUNs" on page 245. ◆ Verify the Database in this Backup This screen appears only if you selected a backup that contains an unverified database. ◆ Advanced Restore Options For a list of information you need to provide, see "Information you need to perform a restore" on page 226. **Result:** After you click Next, the Completing the Backup Wizard screen appears and displays the restore settings you have selected.
10	Go to Step 12.

Step	Action
Restoring from a multiple-database backup set	
11	Follow the instructions in the remainder of the Restore Wizard screens: ◆ Backup to Restore In the left panel, expand the list and then double-click to select the multiple-database backup set from which you want to restore. ◆ Restore Databases and Transaction Logs ❖ By default, all databases and transaction log backups are selected. To deselect or reselect all transaction log backups, click Select All or Unselect All. ❖ If you want to perform a point-in-time restore, select the Point in Time Restore option. ◆ Database State after the Restore For detailed instructions, see "Specifying multiple-database recovery state from within the Restore Wizard" on page 443. ◆ Restore a Database as a New Database ◆ Database Restore Target Location To restore to an alternate location using a writable Snapshot copy, select the "Restore to Alternate Location using Writable SnapShot" option. For more information, see "Destinations for a restore operation" on page 217 and "Databases restored to or mounted at temporary, alternate locations" on page 217. ◆ Verify the Databases in this Backup This screen appears only if you selected a backup that contains an unverified database. ◆ Advanced Restore Options For a list of information you need to provide, see "Information you need to perform a restore" on page 226. **Result:** After you click Next in the Advanced Restore Options screen, the Completing the Backup Wizard screen appears and displays the restore settings you have selected.

Step	Action
Completing the Restore Wizard	
12	After you verify that all the settings in the screen are correct, click Finish. **Result:** The Restore Wizard closes and the Restore Status dialog box appears and displays the Restore Task List, which will be used to show the progress of the restore operation after you start it.
Restore Status	
13	To start the restore operation, click Start Now. **Result:** SnapManager begins to restore your databases from the backup you selected. SnapManager Restore completes each task and checks it off on the list shown in the Restore Task List view. You can switch back and forth between the task check-off list and the progress report using the Switch buttons on either window. If the restore is successful, the Task window shows the check-off list with the tasks completed, and a dialog box reports that the restore was successful. **Note** If Notification is enabled, e-mail is sent and the event is posted to the Windows event log.
14	After the restore is complete, click OK to close the dialog box. **Result:** Your restore is now complete and your SQL Server computer comes back online.
15	After the restore is finished, you can optionally perform a SnapManager backup and verification to verify that your restored database is free of physical-level corruption.

Managing databases mounted at or restored to alternate locations

SnapManager supports the ability to mount a database without restoring it. See the following topics for more information:

- "Mounting and attaching a database" on page 242
- "Detaching a database and dismounting Snapshot LUNs" on page 245

Mounting and attaching a database

You can mount a database without restoring it. The database is mounted at an alternate, temporary location, but the transaction logs are not applied. Because the *data is not current*, this feature can be used to view only the *layout* of the data.

Note

To view the *current data at a temporary location*, you would need to restore the database to a temporary location using a *writable Snapshot copy*, as described in "Destinations for a restore operation" on page 217.

To mount a database without restoring the data, complete the following steps.

Step	Action	
1	In the SnapManager main window, click the Restore tab.	
2	In the left panel of the Restore tab, expand the list and then double-click to select the database that you want to mount.	
3	In the menu bar, select Action > Mount & Attach Db. **Result:** The Mount Snapshot and Attach Database dialog box appears.	
4	You have the option to verify the selected databases.	
	If...	**Then...**
	You do not want to verify the database	Leave the Run DBCC CHECKDB option unselected, then go to Step 7.
	You want to verify the database	Go to Step 5.

Step	Action
5	Select the Run DBCC CHECKDB option.
6	If you want to leave the databases attached after the DBCC CHECKDB command has finished, then also select the Leave database attached after DBCC option. **Note** If you select this option, you will need to explicitly detach and dismount the Snapshot copies after completing your task or test. This is described in "Detaching a database and dismounting Snapshot LUNs" on page 245. If you do not detach the databases and dismount the Snapshot copy LUNs, you might encounter a busy Snapshot when you attempt to delete the Snapshot copy.
7	Click OK to close the message box and proceed with mounting the databases. **Result:** The Restore Status dialog box appears.
8	Click Start Now to proceed with mounting the database. **Result:** SnapManager begins to mount your databases to a temporary, alternate location. SnapManager Restore completes each task and checks it off the list shown in the Restore Task List view. You can switch back and forth between the task check-off list and the progress report using the Switch buttons on either window. If the databases are successfully mounted, the Task window shows the check-off list with the tasks completed, and a dialog box reports that the restore was successful.

Step	Action
9	Click OK to close the message box. **Result:** A database restored to or mounted at a temporary, alternate location is listed in the Backup and Verification tab with the following label: `SnapLUN` If the database name already exists on the server, or if the backup set consists of more than one database, the database is listed using the following naming convention: *databasename*`__Mount`

Detaching a database and dismounting Snapshot LUNs

If a database is restored to or mounted at a temporary, alternate location using a writable Snapshot copy (described in "Restore to a temporary, alternate location" on page 217 and in "Mounting and attaching a database" on page 242), you must detach the database and dismount the Snapshot copy LUNs when you have finished viewing the layout of the data. If you fail to do this, you might encounter a busy Snapshot condition when you attempt to delete the Snapshot copy.

To explicitly detach a database and dismount the Snapshot copy LUNs, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">In the SnapManager main window, click the Backup and Verification tab.</td></tr>
<tr><td>2</td><td colspan="2">In the left panel of the Backup and Verification tab, expand the list and then select the database Snapshot copy that you have explicitly mounted at a temporary location.

The temporarily mounted databases are listed in the Backup and Verification tab with the following label to the right of the database name:

<code>SnapLUN</code></td></tr>
<tr><td rowspan="4">3</td><td colspan="2">Determine which method to use to detach the temporary database and dismount Snapshot copy LUNs.</td></tr>
<tr><td>If...</td><td>Then...</td></tr>
<tr><td>The database is restored to a temporary location in a Read-Only post-restore state or the database is in a loading state</td><td>Go to Step 4.</td></tr>
<tr><td>The database is restored to a temporary location in an Operational or Non-Operational post-restore state</td><td>Go to Step 7.</td></tr>
<tr><td>The database is only mounted at a temporary location</td><td>Go to Step 7.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Attempt to bring the temporary database into read-write mode.
For details, see your Microsoft SQL Server documentation.
<table>
<tr><th>If...</th><th>Then...</th></tr>
<tr><td>You succeed at bringing the temporary database into read-write mode</td><td>Go to Step 7.</td></tr>
<tr><td>You are unable to bring the temporary database into read-write mode</td><td>Go to Step 5.</td></tr>
</table></td></tr>
<tr><td colspan="2">Use the SQL Server backup utility and SnapDrive</td></tr>
<tr><td>5</td><td>Use SQL Server Enterprise Manager or Management Studio to delete the read-only database.</td></tr>
<tr><td>6</td><td>Use SnapDrive to detach the Snapshot copy LUNs. To do this, follow the instructions in SnapDrive Installation and Administration Guide for your version of SnapDrive.
The procedure is now finished.</td></tr>
<tr><td colspan="2">Use the “Detach Database and Dismount Snapshot LUN(s)” function</td></tr>
<tr><td>7</td><td>Click and hold the right mouse button to open the context menu, then select Detach Database and Dismount Snapshot LUN(s).
Note
The Detach Database and Dismount Snapshot LUN(s) option is grayed out and not selectable if the database restored to the alternate location using a writable Snapshot copy (a) was restored in a Read-Only post-restore state or (b) is in the Loading state.
Result: A message box displays a warning that this operation might take some time and asks you to confirm that you want to proceed.</td></tr>
<tr><td>8</td><td>Click OK to close the message box and proceed with the operation.
The procedure is now finished.</td></tr>
</table>

Deleting restored Snapshot copies

To explicitly delete the oldest LUN Snapshot copies created during previous restore operations, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">In the SnapManager main window, go to the menu bar and select Action > Delete Backup.

Result: The Delete Backups dialog box appears.

Note
You can also use the Delete Backups dialog box to explicitly delete backup Snapshot copies. For details, see “Explicit deletion of backup Snapshot copies” on page 146.</td></tr>
<tr><td>2</td><td colspan="2">Select the Delete Snapshot copy of LUNs Created During Restore option.

Result: The Snapshot copies to Delete panel displays options pertaining to deleting restore Snapshot copies.</td></tr>
<tr><td rowspan="3">3</td><td colspan="2">Specify which Snapshot copies you want to delete.</td></tr>
<tr><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Delete only the oldest restore Snapshot copies</td><td>1. Select the Delete Oldest Snapshot copies In Excess Of option

2. Specify how many of the newest restore Snapshot copies you want to preserve.</td></tr>
<tr><td>Delete all the restore Snapshot copies</td><td>Select the Delete All Snapshot copies Created During Restore option.</td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="4">4</td><td colspan="2">You can proceed with the delete operation immediately, or you can view the list of restore Snapshot copies currently selected for deletion.</td></tr>
<tr><td>If you want to...</td><td>Then...</td></tr>
<tr><td>View the list of restore Snapshot copies that would be deleted</td><td>Go to Step 5.</td></tr>
<tr><td>Delete the restore Snapshot copies</td><td>Click Delete.

Result: The restore Snapshot copies identified by your selections are deleted. When the deletion is complete, a message box appears. You have completed this procedure.</td></tr>
<tr><td>5</td><td colspan="2">Click Delete Preview.

Result: The Delete Backup Data Set Preview dialog box appears. After a moment, the dialog box displays a count and list of the restore Snapshot copies identified for deletion.

If you want to view a report, click Show Report.</td></tr>
<tr><td rowspan="3">6</td><td colspan="2">Based on the contents of the preview list, you can either cancel the operation or proceed.</td></tr>
<tr><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Cancel the operation</td><td>1. Click Close to close the Delete Backup Data Set Preview dialog box.

2. Click Close to close the Delete Backups dialog box.</td></tr>
<tr><td>Delete the restore Snapshot copies listed in the preview</td><td>1. Click Close to close the Delete Backup Data Set Preview dialog box.

2. Click Delete on the Delete Backups dialog box.</td></tr>
</table>

Managing SnapManager Operational Reports 11

This section describes the reports that SnapManager automatically creates for every operation performed. The following topics are covered:

- "Understanding the SnapManager Reports tab" on page 250
- "Viewing a SnapManager report" on page 251
- "Printing a SnapManager report" on page 252
- "Deleting SnapManager reports" on page 253

Related topics:

- "Uninstalling SnapManager" on page 58
- "SnapManager report directory options" on page 399

Attention

You must run SnapManager from the system console, not from a Terminal Services client. Do not use Terminal Services for any type of SnapManager administration, because you might miss critical information that is displayed only in pop-up boxes at the system console.

Understanding the SnapManager Reports tab

Use the SnapManager Reports tab to access the operational reports that are automatically created for SnapManager configuration, backup, restore, backup set deletion, and other miscellaneous operations. Each report is a log file that includes step-by-step details of the operation, the final status of the operation, and any error messages encountered during the operation.

The SnapManager Reports tab consists of a *navigation panel* and a *display panel*. The navigation panel contains a tree structure that enables you to navigate the *folders* into which the individual *reports* are organized. Each report is a log file that is named in the format *mm-dd-yyyy_hh.mm.ss* that serves to timestamp the creation of the report. Note that *hh* represents the hour expressed in military time. The display area displays the contents of the selected log file.

The following paragraphs describe the folders that contain the SnapManager reports.

Backup: Contains a log file for every backup set (full database backup or transaction log backup) created by SnapManager.

Config: Contains a log file for each time SnapManager is used to migrate a database from a local disk to a LUN or from one LUN to another.

Delete Snapshot: Contains a log for every backup set deleted for an SQL Server that is configured using SnapManager.

Miscellaneous: Contains logs for miscellaneous SnapManager operations, such as modifying the backup management group of a backup set, detaching the database on a Snapshot-backed LUN, or dismounting a database and detaching underlying Snapshot copy LUNs.

Restore: Contains a log for every restore operation (whether it is a stream-based restore, a copy-based restore, or an online Snapshot restore) performed on an SQL Server that is configured using SnapManager.

Note

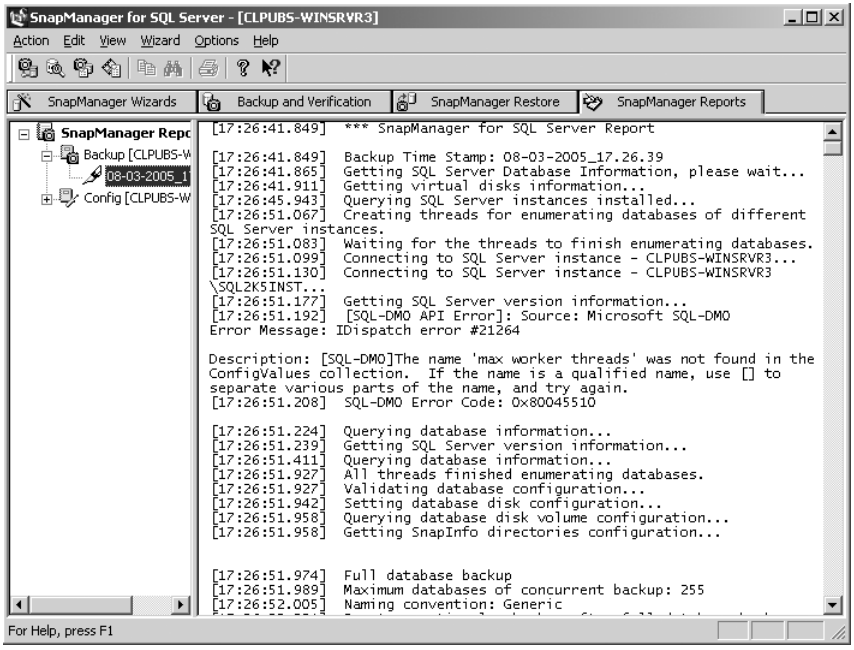
If you are running SnapManager in a Windows cluster, the SnapManager Reports tab could contain additional folders.

Viewing a SnapManager report

To view a SnapManager report, complete the following steps.

Step	Action
1	In the SnapManager main window, click the SnapManager Reports tab.
2	In the navigation panel, click to expand the appropriate reports folder and select the report you want to display in the display panel.

The following illustration shows a report for a SnapManager backup operation.



Printing a SnapManager report

SnapManager provides two ways to print reports, as described in the following sections.

Sending a displayed report to a printer

To send a displayed report to the printer, complete the following steps.

Step	Action
1	In the SnapManager main window, click the SnapManager Reports tab.
2	In the navigation panel, click to expand the appropriate reports folder and select the report you want to print.
3	In the display panel, right-click anywhere within the displayed report to open a context menu, then select Print.

Printing a report using Notepad

To print a report using Notepad, complete the following steps.

Step	Action
1	In the SnapManager main window, click the SnapManager Reports tab.
2	In the navigation panel, click to expand the appropriate reports folder, right-click on the report name to open a context menu, then select Open with Notepad.
3	From within Notepad, go to the menu bar and select File > Print.

Deleting SnapManager reports

To delete an individual report or all reports from a current SnapManager reports folder, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">In the SnapManager main window, click the SnapManager Reports tab.</td></tr>
<tr><td>2</td><td colspan="2">In the navigation panel, click to expand the appropriate reports folder.</td></tr>
<tr><td rowspan="4">3</td><td colspan="2">Specify the report or reports that you want to delete.</td></tr>
<tr><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Delete all the reports in a folder</td><td>Right-click on the report folder to open a context menu, then select Delete All Reports.</td></tr>
<tr><td>Delete an individual report</td><td>Right-click on the individual report name to open a context menu, then select Delete Report.</td></tr>
<tr><td colspan="2">Result: A confirmation dialog box appears.</td></tr>
<tr><td>4</td><td colspan="2">To close the dialog box and proceed with deleting the individual report or all the reports in the folder (as specified), click Yes.</td></tr>
</table>

Replicating Backups to Mirrored Volumes 12

This section discusses how SnapManager can use the Data ONTAP SnapMirror feature to replicate SQL Server database backups to mirrored volumes for data protection and disaster recovery. The following topics are covered:

- "Understanding SnapManager backups with SnapMirror updates" on page 256
- "Minimizing your exposure to loss of data" on page 259
- "Scheduling SnapManager backups with SnapMirror replication" on page 263

Related topics:

- Chapter 9, "Backing Up Databases Using SnapManager," on page 149
- Chapter 14, "Performing Disaster Recovery with SnapManager," on page 291

For more information about SnapMirror: See the documents listed in the following table.

For information about...	See this document...
Using SnapMirror on a storage system: ◆ Data replication using SnapMirror ◆ What mirror volumes are and how they work ◆ Configuring SnapMirror for use with SnapDrive ◆ Creating a SnapMirror destination volume for replicating LUNs	*Data ONTAP Data Protection Guide* for your particular version of Data ONTAP
Using SnapMirror through SnapDrive: ◆ Reasons for using SnapMirror replication ◆ Connecting to a LUN in a SnapMirror destination volume ◆ Configuring the SnapMirror relationship for SnapDrive	*SnapDrive Installation and Administration Guide* for your version of SnapDrive

Understanding SnapManager backups with SnapMirror updates

SnapMirror can be used to replicate SnapManager backups of your SQL Server database to mirrored volumes on a remote storage system. This is described in the following topics:

- "What SnapMirror does" on page 256
- "How SnapManager uses SnapMirror" on page 256
- "Process overview" on page 258

What SnapMirror does

SnapMirror creates replicas of storage system volumes that host LUNs. SnapMirror can asynchronously mirror a Snapshot copy of data on a source volume to one or more volumes configured as destinations of the source volume. The data stored in a mirror on a destination volume can be accessed through SnapDrive.

Note

Because a mirror is volume wide, it includes Snapshot copies of any other data sets contained on the source volume.

SnapMirror updates the destination volume(s) to reflect incremental changes on the source volume. As a result, a destination volume is an online, read-only copy of the source volume at the time of the most recent replication. This data can be used for disaster recovery, offloading tape backup, read-only data distribution, testing on nonproduction storage systems, or online data migration.

A SnapMirror destination volume can reside on the same storage system as the source volume or a different storage system. For disaster recovery purposes, the destination volume generally resides on a different storage system that is also geographically remote from the storage system containing the source volume. For other purposes, the source and destination volumes might exist on the same storage system.

How SnapManager uses SnapMirror

SnapManager backs up your SQL Server data by creating Snapshot copies of the databases and transaction logs. SnapMirror can be used to replicate the volumes that host the Snapshot copies to mirrored volumes on a remote storage system.

SnapMirror replication for SnapManager is volume replication: When you use SnapMirror to replicate SnapManager backups, you can replicate only volumes, not qtrees. SnapManager does not support SnapMirror qtree replication.

Backup Snapshot can trigger SnapMirror updates: SnapManager Backup uses Snapshot copy functionality to back up your SQL Server data to a storage system volume managed by SnapDrive. If the volume has been configured as a SnapMirror source volume with one or more appropriately configured destination volumes, then—upon successful completion of a Snapshot copy backup operation—SnapManager can send a request to SnapDrive to initiate a SnapMirror update of each destination volume.

All three types of SnapManager Backup operations can be configured to trigger SnapMirror updates:

- Full database backup, with or without transaction log backup or database verification
- Transaction log backup only
- Database verification only

Note

The result of the database verification operation (the database integrity status) is written to the backup set, in the SnapInfo directory. By replicating the backup set to the mirrored volume, this status information is kept current on the mirrored volume.

SnapMirror replication for SnapManager is asynchronous: When it is configured directly on the storage system, SnapMirror can be used to perform synchronous replication or asynchronous replication.

When you use SnapMirror to replicate SnapManager backups, however, the replication is performed asynchronously. That is, changes made to the databases between SnapManager backups are not replicated in the SnapMirror destination volume. Therefore, any restore from the destination volume would restore the databases to their state at the time of the last SnapManager backup; subsequent changes to the database that occurred on the source volume after the most recent SnapMirror replication update will not be available if a catastrophic failure occurs before the next update of the SnapMirror destination volume.

A subsequent section, “Minimizing your exposure to loss of data” on page 259, discusses ways to minimize this exposure to data loss.

SnapMirror scheduling: When it is configured directly on the storage system, SnapMirror uses its own replication schedule as configured by a Data ONTAP administrator.

When SnapMirror is used by SnapManager, however, the SnapMirror replication schedule must be disabled on the storage system; SnapMirror updates are instead initiated by SnapDrive upon the completion of a SnapManager backup operation.

A subsequent section, "Scheduling SnapManager backups with SnapMirror replication" on page 263, describes how to do this plus other steps in the process.

Process overview

If your SQL Server databases reside on a storage system volume that is configured as a SnapMirror source volume, then the SnapMirror destination volume is optionally updated after the SnapManager backup operation finishes.

The following sequence provides an overview of how SnapMirror destination replication works:

1. A SnapManager backup is initiated.
2. SnapManager completes all Snapshot copies required for the backup, then requests a SnapMirror volume update through SnapDrive.
3. If any volume whose data is captured in the backup is a SnapMirror source volume, SnapDrive requests information about all SnapMirror destination volumes of that source volume.
4. SnapDrive sends a SnapMirror destination update request to all the related destination volumes.
5. SnapMirror updates the destination volumes to reflect incremental changes to the source volume.

In SnapManager 1.0 for Microsoft SQL Server, a SnapMirror update is initiated for each Snapshot copy as it was taken, resulting in redundant mirror updates. With SnapManager 1.5, SnapManager 2.0 for Microsoft SQL Server, and SnapManager 2.1 for Microsoft SQL Server, the SnapMirror update is delayed until SnapManager explicitly requests it of SnapDrive, after the SnapManager backup is complete.

Minimizing your exposure to loss of data

The following topics discuss methods for minimizing the exposure to loss of data when using automatic SnapMirror updates of SnapManager backup sets:

- "Goal: More frequent mirror updates with minimal Snapshot overhead" on page 259
- "Supplemental replication using rolling Snapshot copies" on page 259
- "Supplemental mirroring of the transaction logs" on page 261

Goal: More frequent mirror updates with minimal Snapshot overhead

As described in "SnapMirror replication for SnapManager is asynchronous" on page 257, changes to the database that occurred on the source volume after the most recent SnapMirror replication update would not be on the destination volume if the source volume were to be lost.

One way to trigger more frequent SnapMirror updates would be to simply use SnapManager to schedule more frequent Snapshot copies of the LUNs containing the transaction logs. Increasing the frequency of SnapManager backup operations, though, increases the difficulty of managing the number of online Snapshot backup sets that are stored online at your primary site. This is described in "Maximum number of Snapshot copies per volume" on page 144.

As an alternative to increasing the frequency of SnapManager backups, you can use SnapDrive to initiate additional, more frequent SnapMirror replication updates. When SnapMirror replication is initiated directly from the SnapDrive command-line, as opposed to using the SnapMirror replication schedule or using SnapManager backups to automatically trigger SnapMirror updates, you can better manage the overhead that is incurred by creating additional Snapshot copies on the primary site.

Supplemental replication using rolling Snapshot copies

When you use SnapDrive to initiate SnapMirror replication, *rolling Snapshot copies* are used. These Snapshot copies are created for the sole purpose of SnapMirror replication. You can use SnapDrive rolling Snapshots to supplement the automatic mirroring of SnapManager backup Snapshot copies with additional, more frequent SnapMirror replication updates of *just the transaction logs*. Exposure to data loss is minimized by narrowing the window during which data can be lost (from the time of the most recently completed SnapManager backup and mirrored Snapshot copy to the time of failure).

Advantages of SnapDrive rolling Snapshots: Using SnapDrive rolling Snapshots to augment your SnapMirror replication schedule for SnapManager offers the following advantages over increasing SnapManager backup operations:

- Fewer Snapshot copies are retained.

  A maximum of two rolling Snapshots are retained at any time.

- Fewer SnapManager backups are required.

  NetApp recommends that you avoid taking SnapManager backups every few minutes, which can result in overlapping SnapManager operations, as well as increased difficulty in managing the large number of resulting backups.

For details about rolling Snapshot, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

Rolling Snapshots of all database files: When all the LUNs utilized by any database file are located in a single storage system volume, replicate *all of the database files* more frequently. This offers the following additional advantages:

- Combined with automatic SnapManager backup set replication, this narrows the window during which transactions can be lost.
- By replicating the entire database, you can quickly recover from site failure by attaching to the database on the remote site.
- When all database files are placed in a single volume, all files will be at the same consistency point. Therefore, when the database is attached, it will automatically recover the most recently committed transactions.

If you augment your SnapManager backup schedule with supplemental SnapMirror replications of all database files, be aware of the following data recovery consideration: if the disaster on the main site has created a suspect database, then the database must be restored from backup, but the transaction log must be backed up using SnapManager before it is restored.

Rolling Snapshots of only the transaction log: In situations where it is not always practical to replicate all database files, replicate *only the transaction log* more frequently (by replicating the LUN where the logs are stored). This offers the following advantages:

- By supplementing SnapMirror updates by using SnapDrive to initiate frequent updates to the transaction log, you can restore the databases up to the point of the most recent successful SnapMirror replication. First, you restore the databases from the replicated backup set on the SnapMirror destination volume. Then you can use the replicated LUN to replay any transaction logs that were generated or changed after that mirrored backup.

- Fewer changes are made to the SQL Server data between replications, as compared to increasing the frequency of mirrored SnapManager backups. This enables you to keep your destination volume effectively more current without running the risk of overlapping your SnapManager operations.

To increase the frequency of the transaction log updates, use the SnapDrive command-line interface tool `sdcli` to schedule the additional SnapMirror replication updates through Windows Scheduled Tasks, specifying supplemental replication of the LUN where the transaction logs are stored.

If you augment your SnapManager backup schedule with supplemental SnapMirror replications of only the transaction log, be aware of the following data recovery considerations:

- The databases must restored from a backup set.
- If the SQL Server on the primary site is available, then SnapManager will request that SQL Server instance to back up the transaction log.
- If the SQL Server on the primary site is not available, then the following activities must be completed before the databases can be restored:
  - The database must be attached.
  - The transaction log must be backed up.

Supplemental mirroring of the transaction logs

If you plan to augment your SnapManager backup schedule with supplemental SnapMirror replications of the transaction logs, your backup schedule requires additional planning.

Allow time for the backup to finish: Design your backup schedule so that a SnapMirror replication is not scheduled to start until the SnapManager backup operation has finished. Note that all three types of SnapManager Backup operations can be configured to trigger SnapMirror updates:

- Full database backup, with or without transaction log backup or database verification
- Transaction log backup only
- Database verification only

At a minimum, be sure to allow enough time for the actual SnapManager backup operation to finish before the SnapMirror update is initiated.

Allow time for the previous replication to finish: Be sure that the interval between SnapMirror replications allows enough time for the previous replication to finish.

Example of a backup and replication schedule: The following table shows an example of a schedule documenting the additional intervals when the volume containing the transaction logs is replicated.

- SnapMirror replication is scheduled to occur 30 minutes after the SnapManager backup begins. You might need to adjust this delay in order to allow enough time for your specific backup configuration to finish.
- It is assumed that the 60-minute interval between each replication of the transaction logs allows enough time for the previous replication to finish.

Regularly scheduled SnapManager backup	Supplemental SnapMirror replication of transaction logs
7:00 a.m.	7:30 a.m. 8:30 a.m. 9:30 a.m. 10:30 a.m. 11:30 a.m. 12:30 p.m.
1:00 p.m.	1:30 p.m. 2:30 p.m. 3:30 p.m. 4:30 p.m. 5:30 p.m. 6:30 p.m. 7:30 p.m. 8:30 p.m. 9:30 p.m.
10:00 p.m.	10:30 p.m.

Scheduling SnapManager backups with SnapMirror replication

When you specify a full database backup operation or a database verification operation for databases that reside on a volume that is a SnapMirror source volume, SnapManager automatically provides you with the option to automatically perform a SnapMirror update after the operation finishes successfully.

Related topics:

- "Creating a full database backup using SnapManager" on page 152
- "Performing database verification using SnapManager" on page 187
- "Scheduling a backup job or a database verification job" on page 349

To schedule a one-time SnapManager backup with SnapMirror replication, complete the following steps.

Step	Action
1	Ensure that SnapDrive is properly licensed for use with SnapMirror. For more information, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
2	Configure SnapMirror on both the source volume to be replicated and its destination volumes. For more information, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

Step	Action
3	Disable the SnapMirror replication schedule on the storage system so that SnapDrive will monitor when a Snapshot copy is taken, and then initiate a replication in response. To do this, modify the schedule parameter (in the `/etc/snapmirror.conf` file or in storage system view mode) to indicate to SnapMirror on the storage system that no schedule is set. Use "- - - -" as the nonschedule time. **Note** It is possible to set the SnapMirror update schedule elsewhere—in the SnapMirror `/etc/snapmirror.conf` file on the storage system. However, when SnapMirror is used by SnapManager, you must rely on the SnapManager backup schedule to drive the SnapMirror replication updates. For more detailed information, see the *SnapDrive Installation and System Administration Guide* for the version of SnapDrive that you are running and the *Data ONTAP Data Protection Guide*.
4	Begin specifying the full database backup operation or the database verification operation. ◆ You can use either the Backup and Verification tab or the Backup Wizard to specify the details of the operation. ◆ Be sure to enable the SnapMirror option. ◆ Be sure to schedule the operation rather than start it immediately. ❖ If you are using the Backup and Verification tab, click Schedule instead of Backup Now or Verify Now. ❖ If you are using the Backup Wizard, deselect the Now option in the When to Run this Operation screen.

Step	Action
5	Use the Schedule dialog box to specify when you want this mirrored backup operation to be run. If you plan to augment this SnapManager backup schedule with supplemental SnapMirror replications of the transaction logs, be sure to follow the guidelines in "Supplemental mirroring of the transaction logs" on page 261. For more information about using the Schedule dialog box in general, see "Scheduling a backup job or a database verification job" on page 349

Archiving SnapManager Backups 13

This section describes how to create offline archives of Snapshot copies containing SnapManager backup sets. The following topics are covered:

- "Understanding SnapManager backup set archival" on page 268
- "Choosing the best way to archive" on page 270
- "Archiving SnapManager backups using NDMP or dump" on page 271
- "Archiving SnapManager backups using a Windows backup utility" on page 274
- "Automatically archiving SnapManager backups using SnapVault" on page 278

Related topics:

- Chapter 9, "Backing Up Databases Using SnapManager," on page 149
- Chapter 14, "Performing Disaster Recovery with SnapManager," on page 291

Understanding SnapManager backup set archival

The following topics provide background information about archiving SnapManager backup sets:

- "Why organizations archive data" on page 268
- "Importance of archiving a complete backup set" on page 268
- "Archiving individual databases is not recommended" on page 269
- "Scheduling SnapManager backups for archiving" on page 269
- "Guidelines for archiving SnapManager backup sets" on page 269

Why organizations archive data

Organizations archive data for many reasons, central of which is disaster recovery. Archiving data enables an organization to create a complete copy of a collection of data suitable for bringing back online at some future date. Whereas backup is concerned with users accidentally destroying files or individual hardware components failing, disaster recovery addresses recovery from events that might disable an entire building or geographical area.

Organizations also archive data for purposes other than disaster recovery. Space constraints often require that older data be archived. Reasons that organizations archive and restore data are as diverse as their businesses. Some organizations restore data for use in historical analysis, and some restore data for use in litigation.

Note

A complete disaster recovery backup strategy must also include system-level backups of the SQL Server.

Importance of archiving a complete backup set

Archived data might be used to completely re-create your SQL Server databases. For this reason, it is imperative that you archive an entire SnapManager backup set. A SnapManager backup set consists of the Snapshot copies of the LUNs that store the SQL Server databases and the SnapInfo directory that is created as part of the backup operation:

- SQL Server database Snapshot copies
- SQL Server transaction log Snapshot copies
- SnapInfo directory Snapshot copy

All of the above components must be archived for you to successfully recover and implement a point-in-time restore.

Archiving individual databases is not recommended

Archiving individual databases from SnapManager backups that contains multiple databases is not recommended. This task requires a full understanding of the Snapshot copy naming conventions used by SnapManager for Microsoft SQL Server and should not be attempted without knowing which Snapshot copies contain the appropriate databases and transaction logs for a given point in time. Archiving complete SQL Server backup sets is recommended.

Scheduling SnapManager backups for archiving

Scheduling SnapManager backups for archiving must take into consideration many factors, including the following:

- Archive method used
- Service Level Agreements for disaster recovery
- Number of SnapManager backups performed per day
- SQL Server client activity schedules
- Backup verification time

Guidelines for archiving SnapManager backup sets

Follow these guidelines when you archive SnapManager backup sets:

- Dedicate your storage system volumes to individual hosts.
- Archive only verified backups. If you are not sure whether a backup is verified, you can use the SnapManager Restore tab to check; a backup with a green check mark is verified.
- Create an archive of the most recent backup. For detailed information, see "SnapManager Snapshot naming conventions" on page 142.
- LUNs cannot be archived using the CIFS or NFS protocols. Use the storage system's `dump` command or an NDMP backup application to archive LUNs.

Choosing the best way to archive

Although all the data required to create an archive is on the storage system, it is not necessarily efficient to back up both of the required archive components using the same backup method. Figuring out exactly how best to tackle the task of archiving depends on the specific environment.

The LUN object that you want to archive is captured in a Snapshot copy located on the storage system. The object can be backed up directly from the storage system using the storage system's `dump` command, backed up directly from the storage system using the NDMP protocol, or even copied out to a NearStore® storage system.

When backing up the specific SnapInfo subdirectory that corresponds to the desired full database backup set, the required data is actually in the active file system of a mounted LUN (which is in the storage system's active file system). There are two ways to back up this particular information.

Using a Windows based backup application, such as the NTBackup utility that ships with Windows, back up the specific directory to a tape device or to a file on a NearStore storage system, for example. Neither an NDMP-based backup or the storage system's `dump` command can back up this data from the SQL Server's active file system.

Another, and much less efficient, method to back up the SnapInfo information is to back up the entire LUN object as captured in a Snapshot copy on the storage system. Doing this is very similar to the method used to back up the LUN object containing the database files. The disadvantage to backing up the LUN object for the SnapInfo directory is that the backup size is that of the entire LUN object itself (the entire LUN—no matter how much or how little data is contained within).

It is important to back up a LUN object that is in a Snapshot copy created by SnapManager. Because LUNs from multiple hosts can be stored on the same storage system volume, only LUNs that belong to the host that created the SnapManager Snapshot copy are consistent. All LUNs within the Snapshot copy that belong to other hosts are not consistent.

Archiving SnapManager backups using NDMP or dump

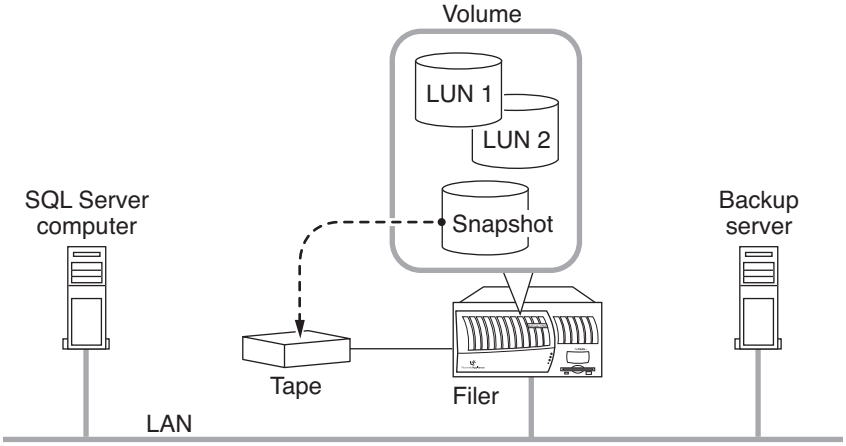
You can use NDMP or the storage system's `dump` command to archive the LUNs containing the SQL Server data and the SnapInfo directory directly from the storage system to the archive medium. NDMP and the `dump` command are the most efficient methods for creating archives of the LUN drive files. See the following topics for more information:

- "About this method" on page 271
- "Guidelines for archiving using NDMP or the dump command" on page 272
- "Procedure for archiving using NDMP or the dump command" on page 273

About this method

When you use NDMP or the storage system's `dump` command to archive your SnapManager backups, you archive each LUN that contains data for that backup set. This method enables you to archive your SnapManager backup sets without involving SQL Server at all. Snapshot copies are taken of the LUNs, then copied to the archive medium and deleted.

This archive method is represented by the following illustration.



Archiving using NDMP or dump

For more information about backing up storage system data to tape, see the *System Administration Data Protection Guide* for your version of Data ONTAP.

Advantages: Using NDMP or `dump` to archive SnapManager backups provides these advantages:

- Because this method does not rely on mounting a Snapshot copy, it does not risk the creation of busy Snapshots.
- Because this method archives the entire raw LUN, restoring is simply a matter of replacing the LUNs.

Note

In this case, you need to run SnapManager in "restore from archive mode" to perform the restore operation. This is described in "Recovering SQL Server databases using archives" on page 309.

- If your archive procedure does not send the data over the network, this method can be significantly faster than other methods.

Disadvantages: The advantages of using NDMP or `dump` to archive SnapManager backups must be weighed against these disadvantages:

- Because you are archiving raw LUNs, the entire LUN containing the SQL Server data is archived, so you archive more data than you need.

Note

If archiving extra data is undesirable, you can use NTBackup or another Windows backup utility to back up the corresponding SnapInfo directory. This must be coordinated so that the two pieces of the archive are kept together for later retrieval.

- If you archive the SnapInfo directory separately, you must ensure that you get both required components from different locations into the same archive.

Guidelines for archiving using NDMP or the dump command

When archiving using NDMP or `dump`, follow these guidelines:

- When using the NDMP- or `dump`-based archive method, back up the database LUN at the following location:

  /`vol`/*VolumeName*/`.snapshot`/`sqlsnap__`*HostName*`__recent`/*FileName*

  `sqlsnap__`*HostName*`__recent` is the name of the Snapshot copy you want to archive.
- Whether the LUN object was archived using the `dump` command on the storage system or through NDMP, backing up the SnapInfo directory can be a separate process. Although the entire LUN where the SnapInfo directory resides can be backed up as a whole, you back up more than you need to; therefore, this method is not very efficient. Backing up the LUN object that contains the SnapInfo directory (this filename is different from the filename

for the LUN that contains the databases) can be done in the same way as backing up the LUN objects for the database disks, as described in the previous two points.

It is more efficient to back up only that which you need, directly from the SQL Server.

Procedure for archiving using NDMP or the dump command

To back up directly from the SQL Server, complete the following steps.

Step	Action
1	Ensure that the LUN is shared. **Note** LUNs are not shared by default.
2	Back up the LUN objects associated with the drive letters that correspond to the database and SnapInfo directory.
3	Back up the LUN object corresponding to the drive letter where the SnapInfo directory is located.

Archiving SnapManager backups using a Windows backup utility

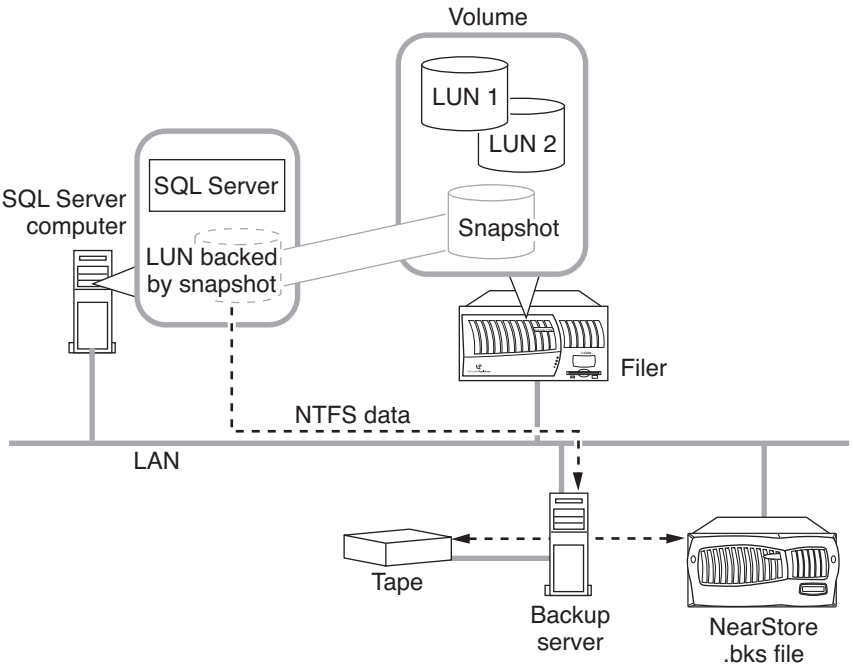
The following topics discuss archiving SnapManager backups using a Windows backup utility:

- "About this method" on page 274
- "Usage example" on page 275

About this method

When you use a Windows backup utility to archive your SnapManager backups, you mount the LUNs backed by the backup Snapshot copy you want to archive, and then use NTBackup or another Windows backup utility to copy the archive data to your archive medium. In this case, the NTFS data is backed up, rather than the raw LUNs.

This archive method is represented by the following illustration.



Archiving using Windows backup utility

Note

The LUN does not need to be mounted on the SQL Server computer; another network host system that is running SnapDrive can be used for this function.

Components that must be included in the archive: The archive must include the following two components:

- The SnapInfo directory as it was backed up directly from the SQL Server computer
- The SQL Server data as extracted from the LUN backed by Snapshot copy

Advantages: Using a Windows backup utility to archive SnapManager backups provides these advantages:

- Because you are archiving NTFS data, not raw LUNs, you can archive exactly the data that you need, and no more.
- The procedures and tools used for this method are probably familiar and available to you.

Disadvantages: The advantages of using a Windows backup utility to archive SnapManager backups must be weighed against these disadvantages:

- Because this method relies on mounting a Snapshot copy, you must be careful to avoid scheduling any backups while the archiving is performed. This is because creating a Snapshot copy of a mounted Snapshot copy results in a Snapshot copy that cannot be deleted. For more information about busy Snapshots, see "Busy Snapshot error prevents deletion of backup set" on page 210.
- You must make sure that you get both required components from different locations into the same archive. Both components must correspond to the same backup set.

Usage example

The following example illustrates one way to use a Windows backup utility to archive your SnapManager backup sets using a script:

1. Mount a Snapshot copy of the LUNs that contain the SQL Server databases
2. Back up the databases using NTBackup.
3. Dismount the LUNs.
4. Back up the SnapInfo directory.

Assumptions: This example is based on the following assumptions, which you might need to alter for your environment:

- The script is run on the computer running SQL Server and SnapManager.
- The drive letter for the SQL Server database is `S:`.
- The Snapshot copy is mounted as drive `V:`.
- The drive letter used for the LUN mount is available when the script is run.

Script: The sample script `SampleScript.txt` contains the following lines:

```
REM Mount a LUN backed by the snapshot of the SQL Server database
REM as drive letter V:\.

sdcli snap mount -k s -s %1 -d v

REM Use NTBackup to back up the database files. The path to the
REM databases in your environment might be different. The /N and /D
REM options are included to name the tape and add a description.

ntbackup backup "V:\Program Files\MSSQL\sqldbdata" /N %1 /D %2

REM Dismount the snapshot mounted as drive letter V:\. The -f
REM option causes the drive letter to be forcefully disconnected.

sdcli snap unmount -d v -f

REM Use NTBackup to back up the SnapInfo directory. The media is
REM appended so as not to overwrite the database backup.

ntbackup backup %3 /T%1 /A
```

Invoking the script and passing parameters: Use the SnapManager Run Command After Operation feature to invoke the script as follows:

```
C:\SnapManager Scripts\SampleScript.txt $SqlSnapshot $SnapInfoName
$SnapInfoPath
```

The three command-line parameters (`$SqlSnapshot`, `$SnapInfoName`, and `$SnapInfoPath`) correspond with the script variables (`%1`, `%2`, and `%3`), as summarized in the following table.

Variable		
Script	**Command-line**	**Sample value passed to the script**
`%1`	`$SqlSnapshot`	`sqlsnap__winsrvr2__recent`

Variable		Sample value passed to the script
Script	Command-line	
%2	$SnapInfoName	WINSRVR2__recent
%3	$SnapInfoPath	I:\SMSQL_SnapInfo\SQL__WINSRVR2\DB__Northwind

For more information about the Run Command After Operation feature, see "Using the Run Command After Operation feature" on page 340.

Automatically archiving SnapManager backups using SnapVault

SnapManager 2.1 for Microsoft SQL Server includes the ability to automatically run a command or script after a backup or verification operation is finished. This feature, called Run Command After Operation, enables you to specify that a particular SnapManager Backup operation is to automatically run a script when the backup is finished. The specification includes any command-line parameters that can be inserted based on the components of the backup or verification. A typical use for this feature, called Run Command After Operation, is to run a script that automatically archives a backup at the time it is created.

You can use SnapVault to automatically archive SnapManager backups sets. See the following topics for more information:

- "About this method" on page 279
- "Planning considerations" on page 279
- "Supporting documents" on page 281
- "Automatically archiving SnapManager backups using SnapVault" on page 281

Related topics:

- "Restoring a storage system volume using backups archived with SnapVault" on page 314
- "Using the Run Command After Operation feature" on page 340

About this method

The following paragraphs summarize how SnapManager backups can be archived using SnapVault.

Use SnapManager to back up the database: Use SnapManager to perform a full database backup and verify the databases in that backup set.

Note

SnapManager cannot be configured to perform database verification directly on the secondary storage. Verification operations must occur for databases residing on the primary storage.

SnapManager launches a script after the backup finishes: The backup operation is specified with the Run Command After Operation option selected.

Note

Only backup sets that have been verified should be archived.

SnapVault archives the backup Snapshot: SnapVault, under the control of the script launched at the end of the SnapManager full database backup operation, is used to transfer the specified backup set from a SnapManager-initiated Snapshot copy on the primary storage system to a SnapVault-managed Snapshot copy on the secondary storage system. SnapVault is used to manage backup retention of the archives on the secondary storage volume.

In the event of a catastrophic failure of your SQL Server environment, SnapVault can be used to recover the LUNs, and then SnapManager can be used to restore your SQL Server databases from these SnapVault archives.

Planning considerations

The following paragraphs describe points to consider when planning your archival scheme.

Determining the frequency of SnapVault updates:

- Any changes to the SQL Server databases that occur after the most recently vaulted backup set cannot be recovered from these archives.
- SnapManager backups created prior to the restore from archive cannot be restored. However, you will be able to restore from other SnapVault archived backups and from any SnapManager backups created after the restore from archive.
- It might not be practical to archive every SnapManager Snapshot copy. Consider archiving only the last backup of the day or the last backup of the week.

Configuring multiple primary storage filers pointing to a single secondary storage system: SnapVault Snapshot copies are queued so that a failure to create one archival Snapshot copy, where multiple primary storage filers point to a single secondary system, causes subsequent archival Snapshot copies not to finish.

By design, SnapVault operates on a volume-wide unit. All the SnapVault qtrees in a secondary volume must follow the same backup schedule, and volume-wide Snapshot copies are coalesced and shared among all the SnapVault qtrees in the volume. SnapVault avoids creating backup/archival Snapshot copies of the configured qtrees when the qtrees are in an unstable or transitioning state, such as when a transfer is in progress.

The backup schedules of the SnapVault qtrees in a secondary volume are inextricably tied. When SnapVault performs backups to all or a subset of these qtrees, some transfers might finish faster than others. However, no further SnapVault updates can begin until the entire group of qtrees is transferred and the data is captured and made visible in a Snapshot copy.

Because SnapVault backup schedules are set on a volume basis, you should aggregate SnapVault qtrees into different volumes, grouped by the desired backup policy and schedule. NetApp recommends that you do not group qtrees with slow data change rates or fast update cycles in the same volume as qtrees with high data change rates or slow update cycles. The reason is that all the SnapVault qtrees in the volume are able to update only as frequently as the qtree that takes the longest amount of time to update.

With the advent of open-systems SnapVault primaries, SnapVault is often used to back up remote offices over the WAN. When the amount of data change spikes, these long-distance updates can take considerably longer to finish, and this condition prevents multiple updates to other qtrees in the volume during that time. To avoid having long-running transfers block updates to other SnapVault secondary qtrees, do not group SnapVault secondary qtrees with vastly different average transfer lengths into the same secondary volume.

Configuring multiple SQL file groups belonging to the same database: Data files for each file group can be located on different LUNs. These LUNs should be located in the same qtree rather than each LUN located in its own qtree. This way, only a single SnapVault relationship needs to be configured for the database as a whole. Configuring each LUN in its own qtree is not recommended because it would necessitate one SnapVault relationship for each qtree. For example, if one SnapVault relationship is unsynchronized or fails for any reason, then the database cannot be restored. SnapManager for SQL Server does not support backup or restore at the file group level.

Supporting documents

Have the following documents available for reference:

- *SnapDrive Installation and Administration Guide* for your version of SnapDrive

  Refer to this document for the following information:

  - Procedure for creating LUNs in qtrees on the SnapVault primary storage system
  - Detailed information about busy Snapshots
- *Data ONTAP Data Protection Guide* for your version of Data ONTAP

  Refer to this document for the following information:

  - Syntax of the various filer commands described in this section
  - Procedure for configuring remote shell (rsh) connection
  - More detailed information about using SnapVault

Automatically archiving SnapManager backups using SnapVault

The procedure for automatically archiving SnapManager backups using SnapVault consists of a sequence of tasks described in the paragraphs that follow:

1. "Configure LUNs on the SnapVault primary storage system" on page 282
2. "Identify the SnapManager backup set you want to archive" on page 283
3. "Configure SnapVault" on page 283
4. "Use SnapManager to trigger updates to the secondary storage" on page 287

The descriptions of these tasks are based on an example scenario in which there are two SQL Server computers sharing the same primary storage. A simple database layout is assumed, where the database and associated log files reside on the same LUN.

The primary storage system is named `filer_pri`, and it contains two storage volumes, `Srv1` and `Srv2`. These storage volumes in turn contain qtrees named `db1` and `db2`, respectively.

The secondary storage system is named `filer_sec`, and it contains a single storage volume, `sv_SQL`.

During the SnapVault update, the two qtrees in the primary storage volumes will be coalesced on the secondary storage volume.

Configure LUNs on the SnapVault primary storage system: Use SnapDrive to create the LUNs in qtrees (rather than directly in the root of the volume) on the primary storage system. You only need to create the qtrees on the primary storage system. Later, when you initialize the SnapVault backup relationship, SnapVault automatically creates the qtrees on the secondary storage system.

Because a single volume on the secondary storage can be the secondary storage for several primary storage systems, the root of the volume on the secondary storage will contain the replicated qtrees from all of the primary storage sources that are configured to use it. When planning the disk configuration of the secondary storage device, minimize any confusion by assigning qtree names that are both descriptive and unique.

Consider using qtree names with the following naming conventions:

- *SQLServerName_DatabaseName*

  Examples:

  ```
  sql_srvr_db1
  sql_srvr_db2
  sql_srvr_db3_and_4.
  ```

- *SQLServerName_InstanceName_DatabaseName*

  Examples:

  ```
  sql_srvr_instance2_db5
  sql_srvr_instance2_db6_and_7
  ```

- *SQLServerName*_snapinfo

  Examples:

  ```
  sql_srvr1_snapinfo
  sq2_srvr1_snapinfo
  ```

Because a SnapVault restore operation restores the entire qtree, it causes any other data that might be on the primary storage system qtree to be overwritten with the data from the SnapVault secondary storage system. For example, if multiple databases share the same LUN, a SnapVault recovery restores both databases.

The same is true if multiple databases are located on their own individual LUNs and share the same qtree. The SnapInfo directory should be on its own LUN, and this LUN should be in its own qtree.

Identify the SnapManager backup set you want to archive: To perform a restore from archive using SnapManager for SQL Server, you need a backup set created by SnapManager. In the context of SnapVault, this means the following:

- The LUN containing the database (or databases) as captured by SnapManager for SQL Server in the Snapshot copies named:

  `sqlsnap__`*SQLServerName_dd-mm-yyyy_hh.mm.ss*

  Example:

  ```
  sqlsnap__sqlsrvr1_11-29-2004_03.16.32
  ```

- The LUN containing the database log files as captured by SnapManager for SQL Server in the Snapshot copies named:

  ```
  sqlsnap__SQLServerName_dd-mm-yyyy_hh.mm.ss
  ```

  Example:

  ```
  sqlsnap__sqlsrvr1_11-29-2004_03.16.32
  ```

- The LUN containing the SnapInfo directory as captured by SnapManager for SQL Server in the Snapshot copy named:

  `sqlinfo__`*SQLServerName_dd-mm-yyyy_hh.mm.ss*

  Example:

  ```
  sqlinfo__sqlsrvr1_11-29-2004_03.16.32
  ```

Note

Because this is a restore-from-archive operation, it is a *point-in-time* restore.

Configure SnapVault: After the qtrees and LUNs have been created on the primary storage system, you can configure SnapVault, as follows:

- Enable SnapVault on both storage systems.
- Initialize the SnapVault relationship and perform a baseline data transfer.
- Create the SnapVault retention policy.

To configure SnapVault, complete the following steps.

Step	Action
Enable SnapVault on both storage systems	
1	Designate one storage system as the SnapVault primary storage and one as the secondary storage, then license each system accordingly using the following filer command: `license add [ sv_ontap_pri | sv_ontap_sec ] =` *LicenseKey* Example command on the primary storage system: `filer_pri>` **`license add sv_ontap_pri=1234567`** Example command on the secondary storage system: `filer_sec>` **`license add sv_ontap_sec=2345678`**
2	Enable SnapVault on both storage systems by using the following filer command: `options snapvault.enable on` Example command on the primary storage system: `filer_pri>` **`options snapvault.enable on`** Example command on the secondary storage system: `filer_sec>` **`options snapvault.enable on`**
3	Configure access between the two storage systems by using the following filer command: `options snapvault.access host=`*SnapVaultFilerName* Example command on the primary storage system: `filer_pri>` **`options snapvault.access host=filer_sec`** Example command on the secondary storage system: `filer_sec>` **`options snapvault.access host=filer_pri`**
4	Designate a volume on the secondary storage system as the destination for the transfers. If necessary, create a new volume by using the storage system's command-line or FilerView.

Step	Action
Initialize the SnapVault relationship and perform a baseline data transfer	

5	This entails the following two tasks:
	◆ Initialize the SnapVault relationship between the two storage systems by specifying the qtrees whose data you want transferred from the primary storage system to the SnapVault secondary storage device.
	Note
	During the baseline transfer, SnapVault automatically creates the qtrees on the secondary storage volume. If you manually created qtrees on the secondary storage volume, delete those qtrees before initializing the relationship; otherwise, the operation fails.
	◆ Trigger the baseline transfer by performing a complete (baseline) transfer of data from the primary storage system to the secondary storage system.
	Both of the preceding tasks are accomplished using the following filer command:
	`snapvault start` [`-k` *kbs*] [`-t` *n*] `-S` *PrimaryFilerName:PrimaryQtreePath* [*sec_host*:] `/vol/`*SecondaryVolumeName/SecondaryQtreePath*
	For a complete description of the command syntax, see the SnapVault section in the *Data ONTAP Data Protection Guide*.
	The following commands exemplify the commands that you would use to initialize the SnapVault relationship in the configuration example:
	`snapvault start -S filer_pri:/vol/Srvr1/db1 filer_sec:/vol/sv_SQL/Srvr1_db1`
	`snapvault start -S filer_pri:/vol/Srvr2/db1 filer_sec:/vol/sv_SQL/Srvr2_db1`
	`snapvault start -S filer_pri:/vol/Srvr1/SnapInfo filer_sec:/vol/sv_SQL/Srvr1_snapinfo`
	`snapvault start -S filer_pri:/vol/Srvr2/SnapInfo filer_sec:/vol/sv_SQL/Srvr2_snapinfo`

Step	Action
Create the SnapVault retention policy	

6	Create the SnapVault retention policy by issuing the following filer command at the storage system console:
	
	`snapvault snap sched` *volume_name snap_name schedule_spec*
	
	For a complete description of the command syntax, see the SnapVault section in the *Data ONTAP Data Protection Guide*.
	
	The following sample commands direct SnapVault to create a retention policy for two different SQL databases that are located on different servers and configured to use the same secondary storage volume named `sv_SQL`:
	
	`snapvault snap sched sv_SQL sqlsnap_srvr1 6@-`
	`snapvault snap sched sv_SQL sqlsnap_srvr2 8@-`
	
	Note that the prefix for the first server is `sqlsnap_srvr1` and the retention level is six Snapshot copies. The prefix for the second server is `sqlsnap_srvr2` and the retention level is eight Snapshot copies.
	
	Specify Snapshot copy names with all lowercase letters. SnapDrive creates Snapshot copy names using lowercase letters and requires that any Snapshot copy you connect to also be in lowercase. If the Snapshot copy name in the retention policy is created with mixed or uppercase letters, SnapDrive will not be able to mount the LUN in this Snapshot copy.
	
	Specify archive schedules as shown in the preceding example. The `@-` indicates that this is a retention policy rather than a schedule of Snapshot copies. The SnapVault updates are driven by SnapManager rather than by the SnapVault Snapshot copy update schedule.
	
	Attention
	When planning the archive schedule and mounting Snapshot copies, use caution to avoid creating busy Snapshots. This is described in "Busy Snapshot error prevents deletion of backup set" on page 210. For more information about busy Snapshots, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

The result is a set of Snapshot copies named as follows, with the newest Snapshot copy having the lowest-numbered suffix:

```
sqlsnap_srvr1.0 <newest>
sqlsnap_srvr1.1
sqlsnap_srvr1.2
sqlsnap_srvr1.3
sqlsnap_srvr1.4
sqlsnap_srvr1.5 <oldest>
sqlsnap_srvr2.0 <newest>
sqlsnap_srvr2.1
sqlsnap_srvr2.2
sqlsnap_srvr2.3
sqlsnap_srvr2.4
sqlsnap_srvr2.5
sqlsnap_srvr2.6
sqlsnap_srvr2.7 <oldest>
```

Use SnapManager to trigger updates to the secondary storage: After the SnapVault relationship is configured and a retention policy defined, you can use SnapManager to trigger SnapVault updates. You can configure SnapManager to run a command or script after a backup and then pass it variables that correspond to the required Snapshot copy names described previously.

To use SnapManager to trigger updates to the secondary storage, complete the following steps.

Step	Action
Create a script to perform SnapVault updates	

1	Create a script that requests SnapVault updates be made to the secondary storage system.
	
	The following sample script requests SnapVault updates for the qtrees in volume `sv_SQL` on the secondary storage and then requests an archival Snapshot copy for the retention policy you set up. In this case, `sqlsnap_srvr1` is the Snapshot copy name prefix corresponding to the retention policy.
	
	`REM Begin sample script`
	`REM Update SnapVault secondary using $SqlSnapshot`
	`rsh filer_sec snapvault update -s %1`
	`filer_sec:/vol/sv_SQL/Srvr1_db1`
	`REM Update SnapVault secondary using $InfoSnapshot`
	`rsh filer_sec snapvault update -s %2`
	`filer_sec:/vol/sv_SQL/Srvr1_snapinfo`
	`REM Create snapshot to archive the above transfers`
	`rsh filer_sec snapvault snap create sv_SQL sqlsnap_srvr1`
	`REM End sample script`
	
	This example shows the complete command-line syntax is shown, but you can simplify the command by omitting the secondary filer's name ("`filer_sec:`") from the secondary qtree path. SnapVault queues up the transfers and subsequent archival Snapshot copy so that the script finishes quickly, even though the transfer might not finish right away. The sample script assumes that the database and log files reside in the same qtree (whether on the same LUN or different LUNs in the same qtree).
	
	Note
	This script requires a remote shell (rsh) connection to the secondary storage system. The SnapManager Server Identity account is the account for which you need to configure rsh privileges. Consult the *Data ONTAP Data Protection Guide* for details on configuring rsh. When scheduling a backup job in SnapManager with the Run Command After Operation option, be sure to specify the same username and password for the scheduled job.

Step	Action
2	If the script is located on a UNC path (a network share) and if SnapManager is running on Windows Server 2003, then be aware of the following limitation: With Windows Server 2003, Enhanced Internet Explorer Security Configuration is enabled by default, preventing SnapManager from running scripts located at a UNC path. If the script is located on a UNC path and SnapManager is running on a Windows Server 2003 system on which Enhanced Internet Explorer Security Configuration is enabled, then SnapManager can run the script only if you add the network location to Internet Explorer's list of trusted sites. For details, see "Running a script from a UNC path on a Windows Server 2003 system" on page 347.
Configure SnapManager to create uniquely named backup Snapshot copies	
3	Configure SnapManager to use a unique Snapshot copy name rather than the generic "`_recent`" name. This causes all database backup Snapshot copy names—even for the most recent backup—to contain the Snapshot copy creation date and time. To do this, use the Backup Settings dialog box and select Use Unique in the "Select a backup naming convention..." panel. For details, see "Configuring the profile of a full database backup" on page 378.
Start or schedule a SnapManager full database backup and verification operation that also launches the SnapVault update script	
4	Start or schedule a full database backup, being sure to select the Run Command After Operation option. **Result:** When you start or schedule the backup operation, the Run Command After Operation dialog box appears.

Step	Action
5	In the Run Command After Operation dialog box, specify the following information: ◆ The name of the host system on which the script resides ◆ The full path to the command or script ◆ The variables that you want to pass to the script Populate the Command Arguments box with the following SnapManager variables in the order indicated: **a.** `$SqlSnapshot` The variable represents the name of the "`sqlsnap__`" Snapshot copy that contains the database and transaction log LUNs. **b.** `$InfoSnapshot` The variable represents the name of the "`sqlinfo__`" Snapshot copy that contains the SnapInfo directory. For details, see "Using the Run Command After Operation feature" on page 340. **Note** The SnapVault updates and the archival Snapshot copies triggered by the script will not finish right away, even though the script itself does. Similarly, a restore operation is not instantaneous. For this reason, it is important to monitor the SnapVault status regularly to track its progress and successful completion.
Monitor the SnapVault status	

6	Understand and use the following commands on the storage system to monitor the SnapVault status:
	`snapvault status`
	`snap list -q`
	`snap list -o`
	These commands are documented in the SnapVault section of the *Data ONTAP Data Protection Guide*.

Performing Disaster Recovery with SnapManager 14

This section describes how to recover from catastrophic loss of an SQL Server in a SnapManager for SQL Server environment. The following topics are covered:

- "Guides you should read and have at hand" on page 292
- "Preparing for disaster recovery" on page 293
- "Backing up your SQL Server environment" on page 295
- "Replicating your SQL Server environment" on page 296
- "Restoring your SQL Server environment" on page 298
- "Recovering SQL Server databases using SnapMirror" on page 301
- "Recovering SQL Server databases using archives" on page 309
- "Restoring a storage system volume using backups archived with SnapVault" on page 314
- "Recovering a failed SQL Server computer" on page 321
- "Recovering both a failed storage system and a failed SQL Server computer" on page 324
- "Restoring databases from other SQL Server backups" on page 326
- "Restoring system databases from SnapManager backup sets" on page 337

Related topics:

- Chapter 12, "Replicating Backups to Mirrored Volumes," on page 255
- Chapter 13, "Archiving SnapManager Backups," on page 267

Guides you should read and have at hand

In addition to this guide, you should be familiar with the following guides and have them at hand, available for reference during disaster recovery.

For information about...	See this document...
Microsoft Windows 2000	Microsoft Windows 2000 documentation
Preparing for loss of an SQL environment in the event of disaster	For Microsoft SQL Server 2000: ◆ *Microsoft SQL Server 2000 Administrator's Companion* ◆ *Microsoft SQL Server 2000 Operations Guide* ◆ Any related Microsoft documentation For Microsoft SQL Server 2005: ◆ *Microsoft SQL Server Books Online* (installed with the application) ◆ *Microsoft SQL Server 2005 Administrator's Companion* ◆ *Microsoft SQL Server 2005 Operations Guide* ◆ Any related Microsoft documentation
Data ONTAP	*Data ONTAP System Administrator's Guide* for your version of Data ONTAP
SnapMirror	*Data ONTAP Data Protection Online Backup and Recovery Guide* for your version of Data ONTAP
Connecting to and using the LUNs	*SnapDrive Installation and Administration Guide* for your version of SnapDrive

Preparing for disaster recovery

The following topics discuss preparation for disaster recovery:

- "Recommendations for disaster recovery preparation" on page 293
- "Components required to recover your SQL Server" on page 293
- "Methods for getting SQL Server data off site" on page 294

Recommendations for disaster recovery preparation

Preparing for disaster ahead of time is key to recovering from it. The first steps to recovery from catastrophic failure of an SQL Server should take place long before the event occurs. Every environment and site is unique, and every company has different requirements that depend on the critical nature of the data stored. Taking these differences into account, consider the following general disaster recovery preparation recommendations:

- Locate the archive media, such as a NearStore system, off site or at a remote location.
- Establish a policy of redundant knowledge by ensuring that more than one person knows how to restore the system.
- Maintain detailed records and change logs and ensure that they are always available.
- Perform mock disaster recovery processes on your nonproduction SQL Servers to become acquainted with real-life situations.

Components required to recover your SQL Server

To be able to reconstruct your SQL Server environment after a catastrophic failure, you recover the following three components:

Windows environment: You can either back up or replicate your Windows environment. For more information about backing up your Windows environment, see "Backing up your Windows environment" on page 295.

SQL Server: Back up your SQL Server. For more information about backing up your SQL Server, see "Backing up your SQL Server" on page 295.

SQL Server databases: You use SnapManager to create backups of your SQL Server data, then move that data off site, either by archiving those backups or using SnapMirror to mirror them to another filer at a remote site.

Methods for getting SQL Server data off site

Choose from among the following methods for getting your SQL Server data offsite in preparation for disaster recovery:

- Use SnapMirror to mirror your storage system data to a storage system in another location. For more information, see Chapter 12, "Replicating Backups to Mirrored Volumes," on page 255.
- Archive the data to physical media, such as tapes, and store that media offsite. For more information, see Chapter 13, "Archiving SnapManager Backups," on page 267.
- Automatically archive SnapManager backups using SnapVault. For more information, see "Automatically archiving SnapManager backups using SnapVault" on page 278.

The following table summarizes advantages and disadvantages of each method.

Method to get SQL Server data offsite	Advantages	Disadvantages
Using SnapMirror	◆ Restoring from a SnapMirror destination is significantly faster than restoring from tape. ◆ The destination can be updated more frequently than by using tape, resulting in more current data.	◆ Requires another storage system in the remote location. ◆ Requires WAN connectivity to the remote location, with sufficient bandwidth. ◆ Mirrors only backups sets that are on the source storage system.
Archiving SnapManager backups to tape	◆ The required equipment might already be available and in use.	◆ Restoring from tape is significantly slower than restoring from a storage system. ◆ Tapes must be stored and managed.
Archiving SnapManager backups using SnapVault	◆ Restoring from a SnapVault archive is faster than archiving from tape. ◆ The destination can be updated more frequently than by using tape. ◆ Backup sets that are no longer available on the primary storage can be retained.	◆ Requires another storage system in the remote location. ◆ Requires WAN connectivity to the remote location, with sufficient bandwidth. ◆ Recovery requires data to be replicated back to the original storage system.

Backing up your SQL Server environment

The following topics describe backing up your SQL Server environment:

- "Backing up your Windows environment" on page 295
- "Backing up your SQL Server" on page 295

Backing up your Windows environment

SnapManager, the SQL Server, and the storage systems are dependent on the Windows environment. Before you can use any of the processes in this section, the Windows environment must be completely restored. Therefore, it is important that you back up your Windows environment so that you can restore the same state as part of the recovery process.

To back up your Windows environment, you must complete, at a minimum, the following high-level process.

Step	Action
1	Back up your SQL Server, including your Windows operating system and any applications running concurrently with the SQL Server.
2	Use your backup utility to create and maintain a current emergency repair disk (ERD).

Backing up your SQL Server

To ensure that you back up all the required components on your SQL Server, follow the process outlined in the appropriate Microsoft document.

For Microsoft SQL Server 2000:

- *Microsoft SQL Server 2000 Administrator's Companion*
- *Microsoft SQL Server 2000 Operations Guide*
- Any related Microsoft documentation

For Microsoft SQL Server 2005:

- *Microsoft SQL Server Books Online* (installed with the application)
- *Microsoft SQL Server 2005 Administrator's Companion*
- *Microsoft SQL Server 2005 Operations Guide*
- Any related Microsoft documentation

Replicating your SQL Server environment

The following topics describe replication of your SQL Server environment:

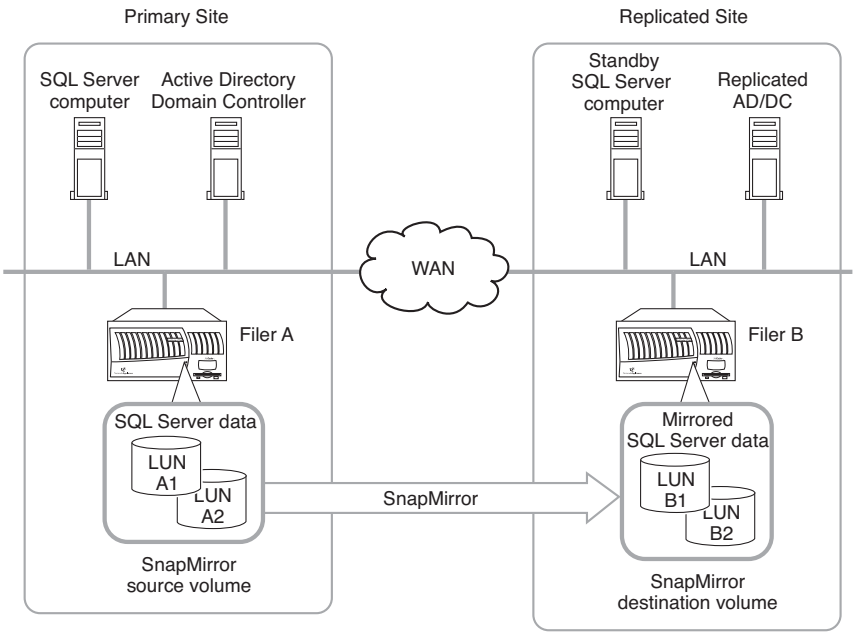
- "Reason to replicate your SQL Server environment" on page 296
- "Example of a replicated site" on page 296

Reason to replicate your SQL Server environment

If you want the ability to recover from a total site outage in a minimum amount of time, you can replicate your SQL Server environment to a remote site. Then, if the primary site is destroyed, you can re-create your SQL Server environment on the replicated site.

Example of a replicated site

The following illustration shows a typical SQL Server site replication.



Example Site Replication

Note the following facts about this site configuration:

- The Windows environment (Active Directory, Domain Controller, and so on) is replicated through the Wide Area Network (WAN) to the replicated site.

For more information about replicating your Windows environment and using a replicated environment to recover from a disaster, see your Windows documentation.

- The SQL Server data on the filer is mirrored using SnapMirror to a filer on the replicated site.

  For more information about setting up SnapMirror, see your *Data ONTAP System Administration Data Protection Guide*.

- The standby SQL Server is configured identically to your primary SQL Server, except that it has a different name than the primary SQL Server.

Restoring your SQL Server environment

The following topics describe the recovery from catastrophic loss of an SQL Server environment that integrates SnapManager and storage systems:

- "SQL Server environment recovery processes" on page 298
- "Rules and guidelines for disaster recovery" on page 298
- "Choosing a recovery procedure" on page 299

SQL Server environment recovery processes

Recovery from catastrophic loss of an SQL Server environment that integrates SnapManager and storage systems is outlined in the following process.

Recovering the Windows environment: All procedures in this document assume that you already recovered your Windows environment.

For more information about recovering your Windows environment, see the Microsoft documentation listed in "Guides you should read and have at hand" on page 292.

Recovering the SQL Server: This process is performed using the backup application you used to back up the SQL Server.

Recovering the SQL Server databases: This process is performed using SnapManager.

Rules and guidelines for disaster recovery

Before you create a disaster recovery plan using SnapManager, be sure that you understand the following rules and guidelines.

Disaster recovery using SnapMirror replication: The following rules and guidelines apply to disaster recovery using SnapMirror replication of SnapManager backups:

- In a SnapMirror configuration, you can use an up-to-the-minute restore operation to restore your databases to the point in time of the last complete backup set that was replicated.
- In order to be able to perform an up-to-the-minute restore in a SnapMirror configuration, you would need to have incorporated supplemental rolling Snapshots in conjunction with more frequent mirroring of transaction logs.

  This is described in "Minimizing your exposure to loss of data" on page 259.

Disaster recovery using archives: The following rules and guidelines apply to disaster recovery using archived Snapshot copies containing SnapManager backup sets:

- Successful disaster recovery from an archive requires that the archive contains an entire SnapManager backup set. A SnapManager backup set consists of the Snapshot copies of the following items:
  - Snapshot copies of the LUNs on which the data files reside
  - Snapshot copies of the LUNs on which the transaction log files reside
  - Snapshot copies of either the related SnapInfo directories or the LUNs on which the related SnapInfo directories reside
- When using SnapManager to restore databases from an archive, the LUN files from the archive must be restored into their original location in the active file system of the storage system's volume.

General rules and guidelines: The following rules and guidelines apply to both types of disaster recovery methods:

- The drive letters assigned to the LUNs that are restored must be the same drive letters that were used when the archive was created.
- You cannot recover a LUN or transaction logs into a read-only file system, such as a Snapshot copy.
- For SnapManager Restore to work properly, SnapManager and SnapRestore must be licensed on the storage system that stores the SQL Server databases.
- The recovery processes in this guide do not address a large-scale disaster in which supporting Windows infrastructure, such as Active Directory and DNS, is damaged or lost. Before attempting recovery of the SQL Server or the storage system, you must recover the supporting infrastructure.

Choosing a recovery procedure

You can use the following table to determine which recovery process most closely matches your needs.

If...	Then use...
You want to recover SQL Server databases from a SnapMirror destination volume	"Recovering SQL Server databases using SnapMirror" on page 301
Your SQL Server computer is online and you want to recover SQL Server databases from archive	"Recovering SQL Server databases using archives" on page 309

If...	Then use...
Your storage system volume has failed or been destroyed and you used SnapVault archiving	"Restoring a storage system volume using backups archived with SnapVault" on page 314
Your SQL Server computer has failed or been destroyed	"Recovering a failed SQL Server computer" on page 321
Both your SQL Server computer and your storage system have failed and you want to recover on the same hardware	"Recovering both a failed storage system and a failed SQL Server computer" on page 324

Recovering SQL Server databases using SnapMirror

After the failure of a storage system or a volume on a storage system, you can recover SQL Server databases that are mirrored using SnapMirror, as described in the following topics:

- "Preparing for disaster recovery using mirrored volumes" on page 301
- "Procedure overview" on page 301
- "Performing disaster recovery using mirrored volumes" on page 303

Preparing for disaster recovery using mirrored volumes

To be able to recover SQL Server databases using SnapMirror, you must complete the following disaster recovery preparation tasks:

Configure SnapMirror to replicate SQL Server database backups to mirrored volumes: You can configure SnapMirror to use a destination volume on the same storage system or a different, remote storage system. Whether the destination volume is on the same or a different storage system as the failed volume, the disaster recovery procedure is largely the same. For more information about configuring the Data ONTAP SnapMirror feature, see *Data ONTAP Data Protection Guide* for your particular version of Data ONTAP.

Note the drive letter mappings: For each LUN on the SnapMirror source volume, note the relevant drive letter mapping on the SQL Server computer. You need to use the same mappings during the disaster recovery procedure when you use SnapDrive to connect to the corresponding LUNs on the SnapMirror destination volume.

Procedure overview

The procedure entails recovering the relevant LUNs—the LUNs containing Snapshot copies of the databases to be restored—from the SnapMirror destination volume.

Basic procedure: First use SnapDrive to connect to the corresponding LUNs in the SnapMirror destination volume. Then use the SQL Server management tool (Enterprise Manager or SQL Server Management Studio) to attach the database located on the SnapMirror destination volume.

If you cannot attach the database on the SnapMirror destination volume and the transaction log files are intact, follow the steps that describe how to minimize data loss by ensuring that SnapManager restore automatically back up the last active transaction log of the database.

If you succeed in recovering the necessary LUNs, you can use SnapManager to restore from the most recent backup set. Some of the details of this phase of the recovery procedure depend on the nature of the storage system or volume failure:

- Only SQL Server database data files were lost
- Only SQL Server transaction log files were lost
- Both SQL Server database files and transaction log files were lost

If the transaction log files were lost: The recovery procedure includes special steps you must take if the transaction files were lost:

- If the *transaction log files* were lost, the *active transactions* were lost and are unrecoverable. Because the active transactions are unavailable, you must use SnapManager to perform a *point-in-time restore and not an up-to-the minute restore*.

  For descriptions of the two types of restore operations, see "Types of SnapManager restore operations" on page 222.

- In addition, the SnapManager Restore from the SnapMirror destination volume must be performed with the *transaction log backup option disabled* if the transaction log files were lost.

  If you fail to disable this transaction log backup, subsequent SnapManager backup sets reside on a recovery path that is inconsistent with that of the database. Such backup sets cannot be applied to the database; attempts to restore the database from such backup sets results in failure, with the following error message in the restore log:

  ```
  Failed with error code 0x800410df
  ```

If this occurs, perform the restore again, but do not apply transaction logs.

Performing disaster recovery using mirrored volumes

To recover SQL Server databases whose backup sets have been mirrored using SnapMirror, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">1</td><td>If...</td><td>Then...</td></tr>
<tr><td>You are recovering SQL Server 2000 databases</td><td>Do the following:
a. Stop the SQL Server.
b. Continue with Step 2.</td></tr>
<tr><td>You are recovering SQL Server 2005 databases</td><td>Go to Step 2.</td></tr>
<tr><td>2</td><td colspan="2">If any LUNs from the failed source volume still appear to be connected, disconnect them.</td></tr>
<tr><td>3</td><td colspan="2">Use SnapDrive to connect to the corresponding LUN or LUNs in the SnapMirror destination volume.
Note
Use the same drive letters for connecting to the mirrored LUNs that were used on the source volume.
Result: For each mirrored volume, SnapDrive breaks the mirror and restores the LUN using the most recent Snapshot copy generated by SnapDrive or SnapManager.</td></tr>
<tr><td>4</td><td colspan="2">Restart SQL Server if it has been stopped.</td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>5</td><td colspan="2">Use SQL Server Enterprise Manager or SQL Server Management Studio to attach the database located on the associated LUNs in the SnapMirror destination volume, as follows:</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You succeeded in attaching the database</td><td>Complete this procedure as described in Step 6.</td></tr>
<tr><td></td><td>The database could not be accessed using SQL Server Enterprise Manager or SQL Server Management Studio</td><td>Complete this procedure as described beginning with Step 10.</td></tr>
<tr><td></td><td>You were unable to attach the database</td><td>Complete this procedure as described beginning with Step 10.</td></tr>
<tr><td colspan="3">If you attached the database on the SnapMirror destination volume...</td></tr>
<tr><td>6</td><td colspan="2">The steps for restoring the database depend on whether the transaction log volume was lost:</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You lost only the data files of the database</td><td>Complete this procedure as described in Step 7.</td></tr>
<tr><td></td><td>If you lost only the transaction log files of the database</td><td>Complete this procedure as described in Step 8.</td></tr>
<tr><td></td><td>If you lost both the data files and the transaction log files of the database</td><td>Complete this procedure as described in Step 8.</td></tr>
</table>

Step	Action
Performing the restore if only the data volume was lost	
7	Run SnapManager and use the newest full database backup to perform either an up-to-the-minute restore or a point-in-time restore: ◆ For an up-to-the-minute restore, SnapManager automatically backs up the last active transaction log before performing the restore. ◆ For a point-in-time restore, select the backup set, a combination of transaction log backups to be restored, or both. For more information, see the following topics: ◆ For information about identifying the most recent full database backup, see "SnapManager backup set names" on page 141. ◆ For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 140 and "Transaction log backup" on page 134. ◆ For information about performing a restore operation, see "Performing a restore operation" on page 225. The procedure is now finished.
Performing the restore if the transaction log volume was lost	
8	Disable the option to back up the transaction log prior to performing the restore: **a.** From the SnapManager menu bar, select Options > Restore Settings. **b.** In the Restore Settings dialog box, deselect the "Create transaction log backup before restore" option. **c.** Click OK. For more information, see "SnapManager restore options" on page 382. The reason you must disable this restore option is that the active transactions were lost due to the failure of the volume containing the transaction log.

Step	Action
9	Run SnapManager and use the newest full database backup to perform a point-in-time restore. **Note** Because the transaction log volume was lost, an up-to-the minute restore is not possible. For a point-in-time restore, select the backup set, a combination of transaction log backups to be restored, or both. For more information, see the following topics: ◆ For information about identifying the most recent full database backup, see "SnapManager backup set names" on page 141. ◆ For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 140 and "Transaction log backup" on page 134. ◆ For information about performing a restore operation, see "Performing a restore operation" on page 225. The procedure is now finished.

Step	Action
If you did not attach the database on the SnapMirror destination volume...	
10	If you cannot attach the database on the SnapMirror destination volume and none of the transaction log files were lost, then—to reduce the loss of data—ensure that the last active transaction log of the database is automatically backed up by SnapManager Restore: ◆ See Microsoft KB article 253817, "HOW TO: Backup the Last Transaction Log When the Master and the Database Files Are Damaged." This article describes how you can back up the currently active transaction log even if the SQL Server 2000 database file is damaged, provided that the transaction log file is still accessible. ◆ Use this same Microsoft KB article as a general guide for gaining access to the last active transaction log of the database on the SnapMirror destination volume. While referring to the steps in that article, observe the following key points: ❖ When you create a similar database that contains the same number of data and transaction log files as the original database on the SnapMirror destination volume, you are creating the database you will be restoring using SnapManager. ❖ Instead of using the SQL Server `Backup Log` command to back up the transaction log (as described in the Microsoft article), proceed to the next step in this procedure. For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 140 and "Transaction log backup" on page 134. **Attention** Do not use SQL Server Enterprise Manager or SQL Server Management Studio to back up the last active transaction log. Due to file formatting differences between SnapManager backups and Enterprise Manager or Management Studio backups, SnapManager cannot perform a restore operation using transaction logs that were backed up using Enterprise Manager or Management Studio. If any of the transaction log files were lost, no workaround is possible and you cannot minimize data loss.

Step	Action
11	Use SnapManager Restore to automatically back up the last active transaction log of the database. **a.** Start the SnapManager application. **b.** Select Options > Restore Settings, and ensure that the "Create transaction log backup before restore" option is enabled. This causes SnapManager Restore to automatically back up the last active transaction log before actually performing the restore portion of the operation. **c.** Use the newest full database backup to perform either an up-to-the-minute restore or a point-in-time restore *to the new database* you created in the previous step. For information about identifying the most recent full database backup, see "SnapManager backup set names" on page 141. For general information about performing a restore operation, see "Performing a restore operation" on page 225.

Recovering SQL Server databases using archives

The following topics describe recovery of SQL Server databases using archives:

- "System prerequisites" on page 309
- "Information needed" on page 309
- "Procedure summary" on page 310
- "Procedure details" on page 310

System prerequisites

In order to restore SQL Server databases from archives, the following prerequisites must be met.

Storage system: The storage system must be up and running and ready for data to be restored.

Backup media: The backup media must be available and ready to be used for restore.

Database: If the database is still mounted, detach it, using SQL Server Enterprise Manager or SQL Server Management Studio.

Windows 2000 Server or Windows Server 2003: You must restore the Windows system and all services required by the SQL Server. SnapManager, the SQL Server, and the storage systems depend on Microsoft Windows infrastructure elements such as DNS and Active Directory.

LUNs: Disconnect the LUNs from the Windows host machines.

Information needed

Before you begin restoring your SQL Server databases from archive, you need the following information.

Backup and restore method: You should be familiar with the backup and restore method you are using for the LUNs and SnapInfo directory (either the storage system `dump` command or an NDMP-based backup). See the product documentation specific to the backup application.

Supporting documentation: Have the supporting documentation for SnapDrive, Data ONTAP, and your backup application available for reference.

LUN drive letters: You need to know the original drive letters used by the LUNs because LUN objects restored from archive must be reconnected using the same drive letters.

Procedure summary

The following steps represent a high-level overview of the restore from archive process:

1. Recover the archived LUNs containing the full backup dataset to the active file system of the storage system.
2. Reconnect the LUNs to the original drive letters.
3. Launch SnapManager for SQL in the restore from archive mode and perform a recovery operation using the most recent backup. To do this, start the application from a command-line using either of the following command-line options:
   - `-restorearchive`
   - `/restorearchive`

   Example:

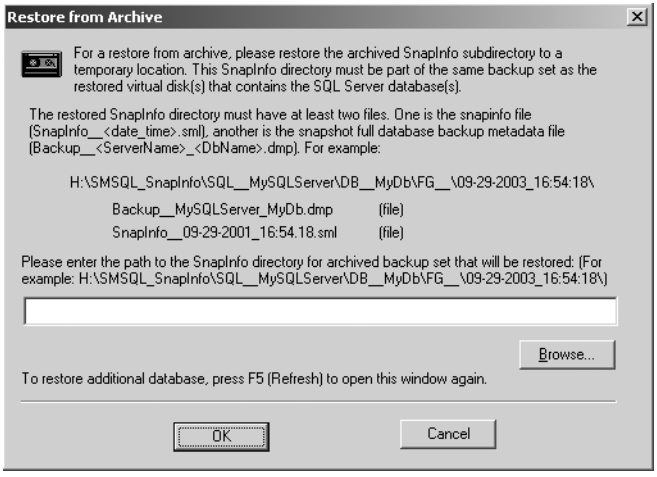
   ```
   C:\Program Files\NetApp\SnapManager for SQL
   Server\SnapManager.exe -restorearchive
   ```

Procedure details

To restore the SQL databases from an archive of a SnapManager backup set, complete the following steps.

Step	Action
1	Using your backup application, restore the LUN object for the databases back into the active file system on the storage system. **Note** If the LUN object containing the SnapInfo directory was backed up (versus only the directory), restore that as well. Consult your backup application's documentation for further details.
2	On the Windows 2000 Server or Windows Server 2003 host system, reconnect the restored LUN objects with the SnapDrive MMC interface using the original drive letters. Consult the SnapDrive documentation for details. Ensure that the LUNs are accessible on the hosting SQL Server.

Step	Action
3	If the specific SnapInfo directory was backed up directly from the active file system of the SQL Server, restore it using your backup application. This directory can be stored anywhere on the Windows host machine or the storage system as long as it is accessible from the SQL Server. **Note** For better protection, restore this information to the active file system on the storage system, however, a share might need to be created and a drive letter mapped so that the directory can be accessed. If the SnapInfo information is backed up as the LUN object, this step can be skipped. See your backup application's documentation for specific details about how to restore data.

Step	Action
4	Launch SnapManager for Microsoft SQL Server from the command-line using the optional command-line switch `-restorearchive` or `/restorearchive`. Example: `C:\Program Files\NetApp\SnapManager for SQL Server\SnapMgrSql.exe -restorearchive` Using this command-line option brings up SnapManager in a mode that enables the restore from archive procedure to work. It allows you to specify a different SnapInfo directory and also prevents a SnapRestore operation. It assumes you restored the LUN yourself. **Results:** Immediately upon launch, SnapManager displays a special Restore from Archive dialog box. 

Step	Action
5	Follow the directions in the Restore from Archive dialog box and browse to the specific SnapInfo subdirectory containing the required data files. **Note** The process only functions properly if the correct directory is selected. Within the directory, you see a few files. Click OK to proceed to the next step. If you accidentally close the Restore from Archive dialog box, you can launch it again by selecting View > Refresh or by pressing the F5 function key on your keyboard.
6	After the correct SnapInfo directory is selected, SnapManager presents you with the standard SnapManager Restore dialog box.
7	Select the most recent backup by double-clicking it.
8	Click Start Restore to begin the point-in-time restore process.
9	Verify the database data updated by the point-in-time restore operation.
10	Exit SnapManager for Microsoft SQL Server.

Restoring a storage system volume using backups archived with SnapVault

The following topics describe restoring a storage system volume from SnapManager backups archived with SnapVault:

- "Procedure details" on page 314
- "Monitoring the SnapVault status" on page 320
- "Multiple databases archived across multiple qtrees in a volume" on page 320

Procedure details

In order to perform a restore, complete the following steps for each qtree you want to restore. You need to do this only for the qtrees where the LUNs containing the database files and transaction logs resided.

Step	Action
1	Using SQL Server Enterprise Manager or SQL Server Management Studio, detach the failed databases to be recovered from archive.
2	If the source LUNs for the failed databases are still online and mapped on the primary storage, do the following: **a.** Note the LUN drive letter assignments. **b.** Unmap the LUNs using FilerView or the `lun` command on the storage system console. **c.** Destroy the LUNs using FilerView or the `lun` command on the storage system console. *Do not delete the SnapInfo LUNs.* **d.** In MSCS configurations, remove any cluster resource dependencies you might have configured on these LUNs.

Step	Action
3	Delete the source qtrees where the LUNs once resided. Qtrees are deleted the same way a regular folder is, using Windows Explorer. The primary storage volume should have a share created on it. You can browse to the share via a UNC path or by mapping a drive letter to it. The qtrees will look like regular folders under the root of the volume (share).
4	Run the Computer Management MMC, then right click on the Disk Management icon and select Rescan Disks. This will clear out any drive letters that might remain from the disconnected LUNs.

Step	Action
5	Perform a SnapVault restore using the `snapvault restore` command, specifying one of the archived Snapshot copies as the source. Restore only the qtrees that contain the databases and transaction logs that you want to restore. **Note** SnapVault restores entire qtrees; therefore, any data in the same qtree as the desired LUN will also be overwritten. Following is an example where the Snapshot copy `sqlsnap_srvr1.0` on the SnapVault secondary storage system is used to recover the qtree db1 on the primary storage system `filer_pri`. `Snapvault restore -s sqlsnap_srvr1.0 -S filer_sec:/vol/sv_SQL/Srvr1_db1 filer_pri:/vol/Srvr1/db1` **Attention** Do not perform a SnapVault restore of the qtree where the SnapInfo LUN resides. Instead, you will mount the LUN in Snapshot copy on the secondary storage system in a subsequent step. The reason for doing this is that performing a restore of the SnapInfo LUN on the primary storage will overwrite all SnapInfo data for all other databases on the SQL Server, whether part of the same backup set or not. This will overwrite newer backup sets for all of the databases on the SQL Server, and you could experience an inability to restore backup sets that were not part of this restore. **Note** You can monitor the status of the recovery transfer by issuing the `snapvault status` command at the storage system console. When the recovery transfer is complete, its status will be listed as "`Broken-off`" by the `snapvault status` command. This does not necessarily indicate that the transfer was successful, only that it is complete. Any errors encountered by SnapVault will be posted directly to the storage system console.

Step	Action
6	Reinitialize the original SnapVault relationship. This can be done by issuing a command similar to command used to create the relationship in earlier steps. `snapvault start -r -S filer_pri:/vol/Srvr1/db1 filer_sec:/vol/sv_SQL/Srvr1_db1` SnapVault does not automatically delete the residual Snapshot copy that was used for the restore, because the copy is required to be in place for this step. It is the only common Snapshot copy and is the basis of restarting the relationship.
7	Delete the residual Snapshot copy left by the SnapVault restore operation. Do this using FilerView or by using the `snap delete` command at the primary storage system console. A residual Snapshot copy is distinguished by the following syntax: `primary host (nvram_id)_vol1_mytree-dst.2`
8	Using FilerView or the `lun` command on the storage system console, unmap the newly restored LUNs.
9	Run the Computer Management MMC, then right click on the Disk Management icon and select Rescan Disks. This will clear out any mappings from the newly restored LUNs. This step is intentionally performed twice in this procedure.
10	Use SnapDrive to reconnect to the newly restored LUNs.

Step	Action
11	Use SnapDrive to connect to the SnapInfo LUN. This LUN needs to be connected on the secondary storage system, using the same Snapshot copy as was used to restore the LUNs in the previous step. Be sure to keep this LUN mounted for as little time as possible. If a Snapshot copy is created on the secondary storage volume while the LUN backed up by the Snapshot copy is mounted, it will create a "busy Snapshot." The chances of this are increased in configurations in which a single secondary storage volume is used to coalesce Snapshot copies from numerous primary storage qtrees. **Note** On Windows Server 2003 Enterprise systems, check to make sure that all of the restored LUNs are writable by creating and then deleting a temporary file or folder. If you find that one or more LUNs are write-protected, use the SnapDrive MMC to disconnect those LUN again. Reconnect to them a second time using the same drive letter.
12	Exit SnapManager if it is running, then restart it from the command-line using the optional command-line switch `-restorearchive` or `/restorearchive`. Example: `C:\Program Files\NetApp\SnapManager for SQL Server\SnapManager.exe -restorearchive` Using this command-line option brings up SnapManager in a mode that enables the restore-from-archive procedure to work.

Step	Action
13	In the Restore from Archive window, enter or browse to the SnapInfo directory corresponding to the latest backup for the first storage group you want to restore. This will be located on the LUN backed up by Snapshot copy on the secondary storage volume, which you connected to in the previous step. If you have multiple-instances or databases to restore from the same backup set, you might need to restore them one at a time. Wait until the first restore finishes, then press the F5 key to open the Restore from Archive window again. This enables you to specify the SnapInfo directory for the next database or instance. Be sure to select the directory corresponding to the most recent backup in that archive set. **Note** If this is a disaster recovery scenario and the LUN on the primary storage system containing the SnapInfo directory was lost, the LUN will need to be re-created using SnapDrive before SnapManager can be used to perform the restore. Create the LUN using the same drive letter that the original LUN had.
14	Perform a point-in-time restore.
15	Verify the database data updated by the point-in-time restore operation.
16	Exit SnapManager.
17	Disconnect the SnapInfo LUN in the Snapshot copy on the secondary storage.
18	Launch SnapManager again, this time without the `-restorearchive` command-line option, and then perform a backup immediately.
19	After a database has been restored from archive, previously created backups will no longer be available. will You need to delete these orphaned SnapManager Snapshot copies manually using FilerView or the storage system console. Ensure that they do not belong to any other backup sets by comparing the time stamps in the Snapshot copy names to the timestamps on backups that might still be valid for any storage groups that were not restored from archive.

Monitoring the SnapVault status

The SnapVault updates and the archival Snapshot copies that are triggered by the script (described in "Use SnapManager to trigger updates to the secondary storage" on page 287) will not finish right away, even though the script itself does. Similarly, a restore operation is not instantaneous. For this reason, it is important to understand and regularly use the following commands on the storage system to monitor the SnapVault status:

```
snapvault status
snap list -q
snap list -o
```

These commands are documented in the SnapVault section of the *Data ONTAP Data Protection Guide*.

Multiple databases archived across multiple qtrees in a volume

When more than one database needs to be restored from archive and not all of the databases were captured in the same archival Snapshot copy on the SnapVault secondary storage volume, you can still perform the restore. This might be the case if SnapManager was configured to back up just one database in a particular backup or if one of several SnapVault qtree relationships in a particular archive Snapshot copy was not up to date.

The way to proceed is to restore one database at a time. Use the preceding instructions to restore the newest qtree you have available corresponding to the database or databases. Then perform a SnapManager "restore from archive" operation. After the first database or databases have been restored, repeat the process for the next set, using the next newest archival Snapshot copy for the restore. Perform another SnapManager "restore from archive" operation.

Recovering a failed SQL Server computer

To recover a failed SQL Server computer, you must use NTBackup or a third-party backup and restore application, relying on its documentation for direction.

Existing backups

This scenario assumes that backups of the SQL Server computer were made; it also assumes that the most recent backup includes the system state of the SQL Server just before the disaster occurred. At a minimum, the following data should be captured on the backup media:

- SQL Server data
  - Any dynamic data on the SQL Server
  - Data that is difficult or impossible to re-create

    Examples include custom scripts, Web pages, and other mission-critical data.
  - Windows backup set: boot partition, system partition, and system state.
- Windows system state
  - Windows 2000 registry
  - Windows 2000 boot files
  - Windows 2000 protected operating system files
- Cluster service registry checkpoints and quorum disk resource data (if you are running cluster service)

Requirements for restoring to a different server

If you are restoring to a different server, that server's hardware must be identical to that of the original server, including the interface cards, hard drives, and firmware versions.

Procedure

To recover a failed SQL Server computer, complete the following steps.

Step	Action	
1	Ensure that the storage system is online and that the volumes containing the LUN objects are available over the CIFS protocol. There must be a CIFS share on the volume on which the LUN resides. See the *Data ONTAP System Administrator's Guide* for your version of Data ONTAP for details about general storage system administration.	
2	Perform a full restore of the SQL Server computer, without the SQL Server databases, using NTBackup or the third-party backup application you used to create the backup. See the documentation for your backup software and the Microsoft disaster recovery documentation for Microsoft SQL Server 2000 or 2005.	
3	**If...**	**Then...**
	Your backup software was configured to back up SnapDrive	Continue with Step 4.
	Your backup software was not configured to back up SnapDrive	Reinstall the same version of SnapDrive used before the disaster occurred. For information about installing and configuring SnapDrive, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
4	Using the SnapDrive MMC, connect the LUNs or ensure that they are connected, and ensure that you are using the same drive letters used before the disaster.	

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">5</td><td>If...</td><td>Then...</td></tr>
<tr><td>Your backup software is configured to back up the entire SQL Server computer less the databases</td><td>Perform a complete recovery of your SQL Server computer using the same backup application. Databases are later restored with SnapManager.</td></tr>
<tr><td>You are not backing up the SQL Server computer other than using SnapManager to back up the databases</td><td>Reinstall the SQL Server software and apply any necessary service packs.</td></tr>
<tr><td>6</td><td colspan="2">Launch SnapManager and run the SnapManager Configuration Wizard to ensure that the correct configuration is used.

If necessary, modify the configuration so that it exactly matches the configuration before the failure.</td></tr>
<tr><td>7</td><td colspan="2">Restore the most recent backup using SnapManager Restore. Do not select the Point-in-Time restore option.

See Chapter 10, “Restoring Databases Using SnapManager,” on page 213.</td></tr>
<tr><td>8</td><td colspan="2">Confirm the operation of the SQL Server.</td></tr>
</table>

Recovering both a failed storage system and a failed SQL Server computer

If both the storage system and the SQL Server computer fail, in most cases you should recover the storage system first so that the data—or the space to recover the data—is available.

Successful recovery of the SQL Server computer depends on the existence of the following components:

- Archives of the SnapManager backup sets containing all the LUN objects

  Note

  For detailed information describing how to prepare for the loss of an SQL Server environment in a disaster, see the *Microsoft SQL Server Administrator's Companion or the Microsoft SQL Server Operations Guide* for your supported Windows operating system.

- Recent, usable backups of the SQL Server 2000 or 2005 databases contained in the restored backup sets

Use the recovery procedure in this section as a guideline for your own recovery plan. For complete information about how to recover an SQL Server and storage system, read Microsoft SQL Server documentation and the appropriate Data ONTAP documentation.

To recover both the storage system and the SQL Server, complete the following steps.

Step	Action
1	Recover the storage system and bring it online. See the *Data ONTAP System Administrator's Guide* for your version of Data ONTAP for information and the *Data ONTAP System Administration Data Protection Guide* for instruction.

Step	Action
2	Unless you are restoring from a tape, perform the following steps: **a.** Install Windows 2000 Server or Windows Server 2003 and load all of the appropriate service packs. **b.** Install SQL Server 2000 or 2005 and load the same service pack that was on the system prior to the crash. **c.** Install SnapDrive and connect to the same drive letters that your LUNs were connected to prior to the crash. **d.** Assuming you backed up your system databases, install SnapManager for SQL Server and migrate your system databases to the same LUNs that they were migrated to prior to the crash.
3	Using SnapManager for SQL Server, recover your SQL Server system databases (master and msdb) from the archived LUN. See "Recovering SQL Server databases using archives" on page 309.
4	Using SnapManager for SQL Server, recover the user databases.

Restoring databases from other SQL Server backups

You can restore databases to the current SQL Server using SnapManager backup sets that were created for a different SQL Server. If the original SQL Server fails, this feature enables you to recover its databases using a different SQL Server.

To use this feature, you must first remap the source LUNs to the current SQL Server, using the same drive letter assignments that were used for the original SQL Server. After you attach the database, use SnapManager to perform either an up-to-the-minute restore or a point-in-time restore of the newest full database backup.

You can perform a restore from other SQL Server backups using the SnapManager Restore tab or the SnapManager Restore Wizard. For detailed information, see one of the following topics:

- "Restoring from other SQL Server backups using the SnapManager Restore tab" on page 326
- "Restoring from other SQL Server backups using the SnapManager Restore Wizard" on page 331

Restoring from other SQL Server backups using the SnapManager Restore tab

To use the SnapManager Restore tab to restore databases to this SQL Server using backup sets created for other SQL Servers, complete the following steps.

Step	Action
1	If the source LUNs for the failed databases are still online and mapped on the primary storage, do the following: **a.** Note the LUN drive letter assignments. **b.** Unmap the LUNs using FilerView or the `lun` command on the storage system console. **c.** In MSCS configurations, remove any cluster resource dependencies you might have configured on these LUNs.
2	On the Windows host system, reconnect the restored LUN objects with the SnapDrive MMC interface, using the original drive letters. Consult the SnapDrive documentation for details. Ensure that the LUNs are accessible on the hosting SQL Server.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>Use SQL Server Enterprise Manager or SQL Server Management Studio to attach the database located on the LUNs.
<table>
<tr><th>If...</th><th>Then...</th></tr>
<tr><td>You succeeded in attaching the database</td><td>Complete this procedure as described beginning with Step 4.</td></tr>
<tr><td>It is not possible to attach the database</td><td>Complete this procedure as described beginning with Step 10.</td></tr>
</table></td></tr>
<tr><td colspan="2">If you attached the database...</td></tr>
<tr><td>4</td><td>Start the SnapManager for Microsoft SQL Server application.</td></tr>
<tr><td>5</td><td>Click to activate the SnapManager Restore tab.</td></tr>
<tr><td>6</td><td>Go to the menu bar and select Action > Restore from Other SQL Server Backups.

Result: The Restore from Backup dialog box appears.</td></tr>
<tr><td>7</td><td>In the Backup Server Name box, select the SQL Server whose backup sets you want to use to restore databases to this SQL Server.</td></tr>
<tr><td>8</td><td>In the SnapInfo Directory Path box, enter or browse to the name of the SnapInfo directory for those backup sets.

Note
Leave the “Use this Server’s SnapInfo Directory” option deselected.

Result: The SnapManager Restore tab is updated to list the backup sets for the other SQL Server you specified.</td></tr>
</table>

Step	Action
9	Use the SnapManager Restore tab to perform an up-to-the-minute restore or a point-in-time restore of the newest full database backup: ◆ For an up-to-the-minute restore, SnapManager automatically backs up the last active transaction log before performing the restore. ◆ For a point-in-time restore, select the backup set, a combination of transaction log backups to be restored, or both. For detailed information, follow the steps in "Restoring using the SnapManager Restore tab" on page 229, continuing at Step 11 on page 233. For background information, see the following topics: ◆ For information about identifying the most recent full database backup, see "SnapManager backup set names" on page 141. ◆ For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 140 and "Transaction log backup" on page 134. The procedure is now finished.

Step	Action
If you did not attach the database...	
10	If you cannot attach the database, then—to reduce the loss of data—ensure that the last active transaction log of the database is automatically backed up by SnapManager Restore: ◆ See Microsoft KB article 253817, "HOW TO: Backup the Last Transaction Log When the Master and the Database Files Are Damaged." This article describes how you can back up the currently active transaction log even if the SQL Server 2000 database file is damaged, provided that the transaction log file is still accessible. ◆ Use this same Microsoft KB article as a general guide for gaining access to the last active transaction log of the database. While referring to the steps in that article, observe the following key points: ❖ When you create a similar database that contains the same number of data and transaction log files as the original database, you are creating the database you will be restoring using SnapManager. ❖ Instead of using the SQL Server `Backup Log` command to back up the transaction log (as described in the Microsoft article), proceed to the next step in this procedure. For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 140 and "Transaction log backup" on page 134. **Attention** Do not use SQL Server Enterprise Manager or SQL Server Management Studio to back up the last active transaction log. Due to file formatting differences between SnapManager backups and Enterprise Manager or Management Studio backups, SnapManager cannot perform a restore operation using transaction logs that were backed up using Enterprise Manager or Management Studio.
11	Start the SnapManager for Microsoft SQL Server application.
12	Click to activate the SnapManager Restore tab.

Step	Action
13	Go to the menu bar and select Action > Restore from Other SQL Server Backups. **Result:** The Restore from Backup dialog box appears.
14	In the Backup Server Name box, select the SQL Server whose backup sets you want to use to restore databases to this SQL Server.
15	In the Snapinfo Directory Path box, enter or browse to the name of the SnapInfo directory for those backup sets. **Note** Leave the "Use this Server's SnapInfo Directory" option deselected. **Result:** The SnapManager Restore tab is updated to list the backup sets for the other SQL Server you specified.
16	Use the SnapManager Restore tab to select the newest full database backup, and then perform an up-to-the-minute restore or a point-in-time restore *to the new database* you created in Step 10. SnapManager Restore will automatically back up the last active transaction log before actually performing the restore portion of the operation. For information about identifying the most recent full database backup, see "SnapManager backup set names" on page 141. For information about performing a restore operation, see "Performing a restore operation" on page 225.

Restoring from other SQL Server backups using the SnapManager Restore Wizard

To use the SnapManager Restore Wizard to restore databases to this SQL Server using backup sets created for other SQL Servers, complete the following steps.

Note

Before you can restore databases to the current SQL Server using backups created for a different SQL Server, you must first remap the source LUNs to this SQL Server, using the same drive letter assignments that were used for the original SQL Server.

Step	Action	
1	If the source LUNs for the failed databases are still online and mapped on the primary storage, do the following: **a.** Note the LUN drive letter assignments. **b.** Unmap the LUNs using FilerView or the `lun` command on the storage system console. **c.** In MSCS configurations, remove any cluster resource dependencies you might have configured on these LUNs.	
2	On the Windows 2000 Server or Windows Server 2003 host system, reconnect the restored LUN objects with the SnapDrive MMC interface using the original drive letters. Consult the SnapDrive documentation for details. Ensure that the LUNs are accessible on the hosting SQL Server.	
3	Use SQL Server Enterprise Manager or SQL Server Management Studio to attach the database located on the LUNs.	
	If...	**Then...**
	You succeeded in attaching the database	Complete this procedure as described beginning with Step 4.
	It is not possible to attach the database	Complete this procedure as described beginning with Step 12.
If you attached the database...		
4	Start the SnapManager for Microsoft SQL Server application.	

Step	Action
5	Make sure that all other Windows Explorer windows are closed on the SQL Server computer running SnapManager.
6	Disable any SnapManager operations that are scheduled to run against the SQL Server data you are restoring, including any jobs scheduled on remote management or remote verification servers.
7	To launch the SnapManager Restore Wizard, go to the SnapManager menu bar and select Wizard > SnapManager Restore. **Result:** The SnapManager Restore Wizard appears and displays the Welcome screen.
8	In the Welcome screen, click Next. **Result:** The Which SQL Server Created the Backups screen appears.
9	In the Which SQL Server Created the Backups screen, do the following: **a.** Select the "Restore SnapManager backups that were created on a different SQL Server" option. **b.** Click Next. **Result:** The SQL Server Where the Backups were Created screen appears.

Step	Action
10	In the SQL Server Where the Backups were Created screen, do the following: **a.** In the Backup Server Name box, select the SQL Server whose backup sets you want to use to restore databases to this SQL Server. **b.** In the SnapInfo Directory Path box, enter or browse to the name of the SnapInfo directory for those backup sets. **c.** Leave the Use this Server's SnapInfo Directory option deselected. **d.** Click Next. **Result:** The Backup to Restore screen appears and lists the backup sets for the other SQL Server you specified.
11	Follow the instructions in the Restore Wizard screens to perform an up-to-the-minute restore or a point-in-time restore of the newest full database backup: ◆ For an up-to-the-minute restore, SnapManager automatically backs up the last active transaction log before performing the restore. ◆ For a point-in-time restore, select the backup set, a combination of transaction log backups to be restored, or both. For detailed information, follow the steps in "Restoring using the SnapManager Restore Wizard" on page 236, continuing at Step 8 on page 238. For background information, see the following topics: ◆ For information about identifying the most recent full database backup, see "SnapManager backup set names" on page 141. ◆ For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 140 and "Transaction log backup" on page 134. The procedure is now finished.

Step	Action
If you did not attach the database...	
12	If you cannot attach the database, then—to reduce the loss of data—ensure that the last active transaction log of the database is automatically backed up by SnapManager Restore: ◆ See Microsoft KB article 253817, "HOW TO: Backup the Last Transaction Log When the Master and the Database Files Are Damaged." This article describes how you can back up the currently active transaction log even if the SQL Server 2000 database file is damaged, provided that the transaction log file is still accessible. ◆ Use this same Microsoft KB article as a general guide for gaining access to the last active transaction log of the database. While referring to the steps in that article, observe the following key points: ❖ When you create a similar database that contains the same number of data and transaction log files as the original database, you are creating the database you will be restoring using SnapManager. ❖ Instead of using the SQL Server `Backup Log` command to back up the transaction log (as described in the Microsoft article), proceed to the next step in this procedure. For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 140 and "Transaction log backup" on page 134. **Attention** Do not use SQL Server Enterprise Manager or SQL Server Management Studio to back up the last active transaction log. Due to file formatting differences between SnapManager backups and Enterprise Manager or Management Studio backups, SnapManager cannot perform a restore operation using transaction logs that were backed up using Enterprise Manager or Management Studio.
13	Start the SnapManager for Microsoft SQL Server application.
14	Make sure that all other Windows Explorer windows are closed on the SQL Server computer running SnapManager.

Step	Action
15	Disable any SnapManager operations that are scheduled to run against the SQL Server data you are restoring, including any jobs scheduled on remote management or remote verification servers.
16	To launch the SnapManager Restore Wizard, go to the SnapManager menu bar and select Wizard > SnapManager Restore. **Result:** The SnapManager Restore Wizard appears and displays the Welcome screen.
17	In the Welcome screen, click Next. **Result:** The Which SQL Server Created the Backups screen appears.
18	In the Which SQL Server Created the Backups screen, do the following: **a.** Select the "Restore SnapManager backups that were created on a different SQL Server" option. **b.** Click Next. **Result:** The SQL Server Where the Backups were Created screen appears.
19	In the SQL Server Where the Backups were Created screen, do the following: **a.** In the Backup Server Name box, select the SQL Server whose backup sets you want to use to restore databases to this SQL Server. **b.** In the SnapInfo Directory Path box, enter or browse to the name of the SnapInfo directory for those backup sets. **c.** Leave the "Use this Server's SnapInfo Directory" option deselected. **d.** Click Next. **Result:** The Backup to Restore screen appears and lists the backup sets for the other SQL Server you specified.

Step	Action
20	Follow the instructions in the Restore Wizard screens to select the newest full database backup, and then perform an up-to-the-minute restore or a point-in-time restore *to the new database* you created in Step 12. SnapManager Restore automatically backs up the last active transaction log before actually performing the restore portion of the operation. For information about identifying the most recent full database backup, see "SnapManager backup set names" on page 141. For information about performing a restore operation, see "Performing a restore operation" on page 225.

Restoring system databases from SnapManager backup sets

The following topics describe the prerequisites and procedures for restoring system databases:

- "Prerequisites for restoring system databases" on page 337
- "Procedures for restoring system databases" on page 337

After the failure of your SQL Server system databases (distribution, master, tempdb, model, and msdb databases), you can restore them from SnapManager backup sets for default and named SQL Server instances.

Prerequisites for restoring system databases

Before you can restore system databases from SnapManager backup sets:

1. The system databases must be migrated to Data ONTAP LUNs. For more information, see "Migrating SQL Server databases to LUNs" on page 107.
2. SnapManager must be used to create stream-based backup sets of those databases. For more information, see "Types of backup operations performed using SnapManager" on page 132.

Procedures for restoring system databases

The procedure to restore your SQL Server system database depends on whether the database is still functional. If the database is no longer functional, you will need to rebuild the system databases first.

Restoring system databases that are still functional: If you are restoring system databases that are still functional, you only need to use SnapManager to restore the system databases from SnapManager backup sets.

For more information, see Chapter 10, "Restoring Databases Using SnapManager," on page 213.

Restoring system databases that are no longer functional: If you are restoring system databases because they are no longer functional, you must first rebuild the system databases using an SQL Server utility:

- For Microsoft SQL Server 2000, use the `Rebuildm.exe` utility to rebuild the system databases. This utility is located in the directory `C:\Program Files\Microsoft SQL Server\80\Tools\Binn`.

  For more information, see your Microsoft SQL Server 2000 documentation.

- For Microsoft SQL Server 2005, use the `setup.exe` utility to rebuild the system databases. This utility is located in the directory `C:\Program Files\Microsoft SQL Server\90\Tools\SetupBootstrap`.

  For more information, see your Microsoft SQL Server 2005 documentation.

To rebuild and then restore SQL Server system databases from SnapManager backup sets, complete the following steps.

Step	Action
1	Create a new Data ONTAP LUN on the same drive letter as the original LUN.
2	Use `Rebuildm.exe` or `setup.exe` to create base system databases. For more information, see your SQL Server 2000 or 2005 documentation. **Result:** The system databases are created in the default location.
3	Migrate the system databases from the default location back to the Data ONTAP LUN. For more information, see "Migrating SQL Server databases to LUNs" on page 107.
4	Use SnapManager to restore the system databases from SnapManager backup sets. For more information, see the following topics: ◆ "Understanding SnapManager Restore" on page 216. ◆ "Types of SnapManager restore operations" on page 222. ◆ "Performing a restore operation" on page 225.

Tools for Managing Backup and Verification A

This section describes SnapManager tools that can be used when managing backup and database verification operations and that can only be accessed within the context of a backup or a database verification operation. The following topics are covered:

- "Using the Run Command After Operation feature" on page 340
- "Scheduling a backup job or a database verification job" on page 349

Note

Other SnapManager application settings can be configured or changed any time after SnapManager has been installed. These settings are described in a different section.

Related topics:

- Chapter 8, "Understanding SnapManager Backup Sets," on page 129
- Chapter 9, "Backing Up Databases Using SnapManager," on page 149

Using the Run Command After Operation feature

SnapManager 2.1 for Microsoft SQL Server includes the ability to automatically run your own program or script after a backup or database verification operation finishes. This feature, called Run Command After Operation, is typically used to automatically archive a backup. See the following topics for more information:

- "How you can launch your own program or script" on page 340
- "Command arguments pass information to your program or script" on page 341
- "Command arguments are operation specific" on page 342
- "How to open the Run Command After Operation dialog box" on page 344
- "Using the Run Command After Operation dialog box" on page 345
- "Running a script from a UNC path on a Windows Server 2003 system" on page 347

How you can launch your own program or script

When you start a SnapManager backup or database verification operation, you have the option to automatically run a command after the operation is finished. If you choose this option, you will be prompted to specify the following information before the backup or database verification operation can begin:

1. The host system from which the command is to be run
2. The full path of the command that you want SnapManager to run after the backup or database verification is finished
3. Any parameters that are to be passed to the command

   Because the command (your own program or script) is invoked from within the context of a specific backup or database verification, you can pass the command information about the components of that operation. In your script, any occurrence of the text string `%1` corresponds to the first parameter passed; the text string `%2` corresponds to the second parameter, and so on.

After you have finished specifying the command and parameters, you can start the backup or database verification operation.

Note

The command or script is run only after a successful backup or verification. If the backup is not completed successfully, or if the verification fails, the command or script is not run.

Command arguments pass information to your program or script

The Run Command After Operation feature supports the following variables, which can pass operation-specific information to your program or script.

Variable	Description

`$SqlSnapshot`	Expands to the name of an SQL Server database Snapshot copy.
	Examples:
	`sqlsnap__winsrvr2__01-31-2005_15.03.09`
	`sqlsnap__winsrvr2__recent`
	Note
	The number of database Snapshot copies in a SnapManager backup set depends on the number of volumes used to store the databases included in the backup.

`$InfoSnapshot`	Expands to the name of a SnapInfo directory Snapshot copy.
	Examples:
	`sqlinfo__winsrvr2__01-31-2005_15.03.09`
	`sqlinfo__winsrvr2__recent`

`$SnapInfoName`	Expands to the name of the SnapInfo directory.
	Examples:
	`WINSRVR2__recent`
	`WINSRVR2_11-23-2004_16.21.07__Daily`
	Note
	If you use this variable, you must also provide the correct path to the directory.

`$SnapInfoPath`	Expands to the name of the SnapInfo subdirectory. Example: `I:\SMSQL_SnapInfo\SQL__WINSRVR2\DB__Northwind`

Variable	Description
$LogBackupFile	Expands to the full path name of the transaction log backup file. Example: I:\SMSQL_SnapInfo\SQL__WINSRVR2\DB__Northwind\LogBackup\ 11-01-2004_13.34.59__Northwind.TRB

Note

Both the $SnapInfoPath and $LogBackupFile variables are automatically enclosed within double quotes so that the actual path name can contain spaces without affecting the script invocation on the Windows command-line. If you do not want the double quotes to appear in your command-line, remove them from the Command Arguments field in the Run Command After Operation window.

Command arguments are operation specific

Each SnapManager operation that supports the Run Command After Operation feature (full backup, transaction log backup, and database verification) parses only the variables that apply to the operation as you have specified it.

The following table shows which of the command variables are available to the Run Command After Operation feature, depending on which SnapManager operation is used to invoke the feature.

Variable	SnapManager operation that is used to invoke the Run Command After Operation feature		
	Full backup[1]	Transaction log backup	Verification of full backup[2]
$SqlSnapshot	Parsed	—	Parsed
$InfoSnapshot	Parsed	Parsed	—
$SnapInfoName	Parsed	—	Parsed
$SnapInfoPath	Parsed	Parsed	Parsed
$LogBackupFile	Parsed[3]	Parsed	—

1. Full backup: In the case of multiple databases, repeat the arguments passed for each database included in the backup.
2. Verification of full backup: In the case of multiple databases, repeat the arguments passed for each database included in the verification. Additionally, variables are parsed and replaced with the most recent backup that is verified as part of this job.
3. Full backup with the Run Transaction Log Backup option selected: The `$LogBackupFile` variable is parsed only when the transaction logs are backed up after full backup.

How to open the Run Command After Operation dialog box

There are two ways to open the Run Command After Operation dialog box while starting or scheduling a backup or database verification operation.

From the Backup and Verification tab: To open the Run Command After Operation dialog box while specifying an operation in the Backup and Verification tab, complete the following steps.

Step	Action
1	Specify a full database backup, a transaction log backup, or a database verification.
2	Select the Run Command After Operation option.
3	Click Backup Now or Verify Now. **Result:** The Run Command After Operation dialog box appears and displays the current default settings.

From the Backup Wizard: To open the Run Command After Operation dialog box while specifying an operation in the Backup Wizard, complete the following steps.

Step	Action
1	Step through the wizard screens to specify a full database backup, a transaction log only backup, or a database verification.
2	In the Run a Command After the Operation screen, do the following: **a.** Select the option labeled "Yes, run a command after this operation." **b.** Click Next. **Result:** The Run Command After Operation dialog box appears and displays the current default settings.

Note

If you want to simply change the default values specified in the Run Command After Operation dialog box—without starting or scheduling a database backup, transaction log only backup, or database verification—you can open the Run Command After Operation dialog box from the Option menu. This is described in "Run Command After Operation default settings" on page 385.

Using the Run Command After Operation dialog box

From within the context of a SnapManager backup or verification operation, you can use the Run Command After Operation dialog box to do the following:

- Specify the details of the command:
  - The computer where you want to run the command (your own program or script)
  - The full path to the command
  - The sequence of SnapManager variables that you want to pass to the command
- Specify whether you want to save the current settings as the default settings

Note

If you want to simply change the default values specified in the Run Command After Operation dialog box—without starting or scheduling a database backup, transaction log only backup, or database verification—you can open the Run Command After Operation dialog box from the Option menu. This is described in "Run Command After Operation default settings" on page 385.

To use the Run Command After Operation dialog box, complete the following steps.

Step	Action
1	Open the Run Command After Operation dialog box. For details, see "How to open the Run Command After Operation dialog box" on page 344.
2	In the "Specify a computer where..." box, enter or browse to the name of the host on which your program or script resides.
3	In the "Specify the full path..." box, browse to your program or script.

Step	Action
4	Form the command input string in the Command Arguments box. You can do this using any combination of the following methods: ◆ To enter text directly into the Command Arguments box, click in the box and type the desired text. ◆ To enter a SnapManager variable into the Command Arguments box, do the following: **a.** If necessary, click in the Command Arguments box to position the cursor. **b.** In the SnapManager Variables list, select the variable you want to enter. For more information, see "Command arguments pass information to your program or script" on page 341 and "Command arguments are operation specific" on page 342. **c.** Click Select. **Note** Both the `$SnapInfoPath` and `$LogBackupFile` variables are automatically enclosed within double quotes so that the actual path name can contain spaces without affecting the script invocation on the Windows command-line. If you do not want the double quotes to appear in your command-line, remove them from the Command Arguments field in the Run Command After Operation dialog box.
5	Repeat Step 4 as needed until the Command Arguments box contains the arguments you want to pass to your program or script.
6	If you want to save the current settings as the new default values, select the Save as default option.
7	Click OK to apply your changes and close the Run Command After Operation dialog box.

Running a script from a UNC path on a Windows Server 2003 system

If SnapManager is installed on a Windows Server 2003 system, then attempting to launch a script from a UNC (Uniform Naming Convention) path might cause SnapManager to hang. If this occurs, the log file shows no indication of the failure other than it is missing the last line of text: "The specified command has been launched successfully."

Windows Server 2003 ships with Internet Explorer's Enhanced Internet Explorer Security Configuration enabled. This setting is restrictive and prevents batch files located on a network share from running. The network location will need to be added to Internet Explorer's list of trusted sites.

To run a script from a UNC path on a Windows Server 2003 system, complete the following steps.

Step	Action
1	Log in to the Windows Server 2003 system that will be running the script. ◆ This system can be the same system that is running SnapManager, or it can be a different server. ◆ You must log on using the same user account that SnapManager is configured to use. For more information, see "SnapManager service account requirements" on page 26.
2	Launch Internet Explorer.
3	In the menu bar, select Tools > Internet Options. **Result:** The Internet Options dialog box appears.
4	In the Internet Options dialog box, do the following: **a.** Select the Security tab. **b.** In the Security tab, select the Local intranet icon, and then click Sites. **Result:** The Local Intranet dialog box appears.

Step	Action
5	In the Local Intranet dialog box, do the following: **a.** In the “Add this Web site to the zone" box, enter the host name of the machine from where the script will be launched. **b.** Click Add. **Result:** The host name is added to the Web Sites list.
6	Click Close to close the Local Intranet dialog box.
7	Click OK to close the Internet Options dialog box.
8	To verify your changes, browse to the network location where the script resides and launch the script. **Result:** The script should now run normally without security prompts. **Note** Although the script should run, it might actually fail to complete properly because the SnapManager variables are not be passed to the script in this test.

Scheduling a backup job or a database verification job

When you specify a SnapManager backup or SnapManager verification operation, you can start the operation immediately or you can schedule the operation to run later. This topic describes how to schedule a backup or verification job for a later time. See the following topics for more information:

- "Choosing a schedule service" on page 349
- "Using the Schedule Job dialog box" on page 349

Related topics:

- "Creating a full database backup using SnapManager" on page 152
- "Creating a transaction log backup using SnapManager" on page 172
- "Performing database verification using SnapManager" on page 187

Choosing a schedule service

You can use either of the following schedule services to schedule a backup or database verification:

- SQL Server Agent
- Windows Scheduled Task Wizard

Note

Some limitations apply to using SQL Server authentication as the security authentication method to schedule the job. For more information, see "About SQL Server authentication" on page 366.

Using the Schedule Job dialog box

Use the Schedule Job dialog box to schedule a backup operation or a database verification operation. The Schedule Job dialog box opens automatically when you finish specifying the backup or verification operation if you have chosen the option to schedule the operation to be run later.

To use the Schedule Job dialog box, complete the following steps from either the Backup and Verification tab or from the Backup Wizard after you have specified the details of the backup or verification operation.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Click Schedule.</td></tr>
<tr><td>2</td><td>If the Run Command After Operation dialog box appears, specify the command and then Click OK to close the dialog box.

For more information, see “Using the Run Command After Operation dialog box” on page 345.

Result: The Schedule Job dialog box appears.</td></tr>
<tr><td>3</td><td>In the Schedule Job Name box, enter a name for your backup job.</td></tr>
<tr><td>4</td><td>Specify what you want to do if a scheduled job of the same name already exists.

<table><tr><th>If you want to...</th><th>Then...</th></tr><tr><td>Overwrite the existing job with this one</td><td>Select the Replace Job if Exists option.</td></tr><tr><td>Be prompted to specify a different name</td><td>Do not select the Replace Job if Exists option.</td></tr></table></td></tr>
<tr><td>5</td><td>In the Select the Scheduling Service to Create Job panel, select the schedule service you want to use.

Note
If you select SQL Server Agent and the service is stopped, SnapManager will automatically start the SQL Server Agent service for you.

<table><tr><th>If...</th><th>Then...</th></tr><tr><td>You want to use the SQL Server Agent</td><td>Select the Replace Job if Exists option, then proceed to Step 6.</td></tr><tr><td>You want to use the Windows Scheduled Task Wizard</td><td>Select the Windows Scheduled Task option, then proceed to Step 10.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td colspan="3">Using the SQL Server Agent</td></tr>
<tr><td>6</td><td colspan="2">In the Server Name box, specify the name of the SQL Server instance that you want to use to run this job.</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You know the server name</td><td>Click in the Server Name box and enter the server host name.</td></tr>
<tr><td></td><td>You prefer to browse to the server name</td><td>Click Browse to use a browse dialog box to select the server host name.</td></tr>
<tr><td>7</td><td colspan="2">Click OK.

Result: The Properties dialog box appears for the job you are specifying. This is an SQL Server Agent dialog box.</td></tr>
<tr><td>8</td><td colspan="2">In the Properties dialog box, specify the parameters of your job schedule:
◆ When the job is to run
◆ If you want the job to repeat, at what frequency</td></tr>
<tr><td>9</td><td colspan="2">Click OK to close the Properties dialog box.

Result: The backup job will run at the times you specified in the Properties dialog box. The backup scheduling process is complete.</td></tr>
<tr><td colspan="3">Using the Windows Scheduled Task Wizard</td></tr>
<tr><td>10</td><td colspan="2">In the Run As box, specify the SnapManager service account that you want to use to run the Windows Scheduled Task Wizard for this job.</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You know the domain and user name</td><td>Click in the Run As box and enter the account name.</td></tr>
<tr><td></td><td>You prefer to browse to the account name</td><td>Click Browse to use a browse dialog box to select the account name.</td></tr>
<tr><td>11</td><td colspan="2">In the Password box and again in the Confirm password box, enter the password for this user.</td></tr>
</table>

Step	Action
12	Click OK. **Result:** The Schedule Job appears with the Schedule tab selected. This is a Windows dialog box.
13	In the Windows dialog box, specify the parameters of your job schedule: ◆ When the job is to run ◆ If you want the job to repeat, at what frequency For information about using this Windows dialog box, see your Microsoft Windows documentation.
14	Click OK. **Result:** The backup job will run at the times you specified in the Schedule Job window. The backup scheduling process is complete. **Note** You can use the Control Panel to modify the schedule or cancel the scheduled job.

SnapManager Command-Line Reference B

This section describes the SnapManager 2.1 for Microsoft SQL Server backup and verification operations you can execute using `SmSqlBi`, the SnapManager command-line utility. The following topics are covered:

- "Guidelines for using the command-line utility" on page 354
- "SmSqlBi syntax" on page 356

Guidelines for using the command-line utility

`SmSqlBi` enables you to enter SnapManager backup and verification commands individually or through automation scripts.

Location of the command-line utility

The `SmSqlBi` command-line utility is located in the SnapManager installation directory. The default installation directory for SnapManager 2.1 for Microsoft SQL Server is as follows:

```
C:\Program Files\NetAppSnapManager for SQL Server\
```

Summary of parameters

When you launch `SmSqlBi`, it displays a summary of parameters as shown below.

```
Usage: SmSqlBi -H <Sql Server host name> [-U <Login> -P <Password>]
-S <SQL Server Name> -C <Number of databases> <Database Name>..
-D <Number of backups to keep>
-R <Retain up-to-the-minute restore ability for older backups in
other management groups>
-K <Number of SnapInfo Drive Snapshots to keep>
[-Q <Verification Server> [-I <verify Login> -W <verify Passwd>]]
[-mp[-mpdir mount_point_directory_path]]
-M <Number of backups to verify>
-A <Attach database after verification>
-RS <Run Command Server>
-RP <Run Command Path>
-RA <Run Command Arguments>
-X DBCC before backup
-Y DBCC after backup
-L Transaction log backup
-G Transaction log backup after full backup
-J Create Snapshot of SnapInfo Drive after backup
-O Remove inactive entries from transaction logs
-N No verification after backup
-Daily Daily management group
-Weekly Weekly management group
-Monthly Monthly management group
-All All management group
-UN Use "Time Stamp" naming convention
-UM Update SnapMirror after operation
```

Tips for using the command-line utility

Observe the following guidelines when using the SnapManager command-line feature:

- All parameters and options are case-insensitive. For example, using the option `-Daily` will achieve the same results as using `daily`.
- Some of the options must be invoked in a particular order. For best results, use the order given in the synopsis for all options.
- When a parameter value string contains spaces, be sure to enclose it in double quotes. For example, use `"First Backup Set"` rather than `First Backup Set`.
- When a parameter has a variable number of input values, it begins with a "count" value that you use to indicate how many values will be included in the parameter. If you specify a number different from the number of values included in the parameter, the command fails.

Note

When you use the SmsqlBi command-line utility with the switches `-MPDIR[Mount Point Directory]` and `-MP,` the corresponding values are set in the registry editor of the server. SnapManager for SQL uses the mount point directory that is specified to mount the Snapshot copy for verification. When you use the SmsqlBi command-line utility without the switches `-MPDIR[Mount Point Directory]` and `-MP,` the default settings are taken from the registry editor for verification.

SmSqlBi syntax

Name

SmSqlBi—Command for accessing SnapManager backup functionality.

Description

This command enables you to initiate a backup or verification job, with all of the options available through the Backup Wizard or Backup and Verification tab in the SnapManager graphical user interface. This feature enables you to create scripts to invoke SnapManager functionality without having to use the SnapManager graphical user interface.

Synopsis

`SmSqlBi -H` *SqlServerHostName* [**`-U`** *UserName* **`-P`** *Password*]
`-S` *Instance* **`-C`** *NumberOfDatabases DatabaseNameList*
[**`-S`** *Instance* **`-C`** *NumberOfDatabases DatabaseNameList*] [*OptionsList*]

Required parameters

The following parameters are required.

Switch	Description
`-H`	You must use the following option to specify the SQL Server computer on which the SQL Server instances reside: **`-H`** *SqlServerHostName* [**`-U`** *UserName* **`-P`** *Password*] *SqlServerHostName* is the host name of the computer on which the SQL Server instances are running. *UserName* is the SQL Server account name (specified if the SQL Server computer is accessed using a different account from that used to access the production SQL Server). *Password* is the SQL Server account password (specified if the SQL Server computer is accessed using a different account from that used to access the production SQL Server).

Switch	Description
-s	You must use the following option to specify the databases or transaction logs you want to back up or verify: **-s** *Instance* **-c** *NumberOfDatabases* [*DatabaseList*] *Instance* is the host name of the SQL Server instance that owns the databases or transaction logs you want to back up or verify. *NumberOfDatabases* is the number of databases to be backed up or verified. If you specify 0, then *all* databases that have valid configurations in that SQL Server instance will be backed up or verified and the database Snapshot names do not need to be specified. *DatabaseNameList* is the list of database Snapshot copy names. The names are separated by single spaces. Specify this list only if you are selecting specific databases in the SQL Server instance (that is, if you are specifying a value greater than zero for the parameter *NumberOfDatabases*). **Note** Repeat this option as many times as needed to specify all databases or transaction logs you want to back up or verify.

Optional parameters

The *OptionsList* shown in the synopsis for this command consists of any valid combination of the following parameters.

Option	Description
-A	If the operation includes database or transaction log verification, use this option if you want to specify that the databases are to be attached after the verification finishes.
-All	If the operation includes database or transaction log verification, use this option if you want to specify that the operation is to include all backup management groups (Daily and Weekly). Compare with the **-Daily** and **-Weekly** options.
-D	If the operation includes database or transaction log backup, use the following option if you want to delete the oldest database backups or transaction log backups: **-D** *NumberOfBackups* *NumberOfBackups* is the number of database backups or transaction log backups to be retained after the oldest backups are deleted.
-Daily	If the operation includes database or transaction log verification, use this option if you want to specify that you only want to verify the backups assigned to the Daily backup management group. Compare with the **-Weekly**, and **-All** options.
-G	If the operation is full database backup, use this option if you want to specify that you also want the transaction logs to be backed up after the database backups finish. **Note** If you want to back up transaction logs only, use the **-L** option instead.

Option	Description
-J	If the operation is a backup of the transaction log only (that is, if the **-L** option is specified), use this option if you want to create a Snapshot copy of the SnapInfo directory after the backup finishes. If you use this option with an operation that includes transaction log backup, you can also use the **-K** option (delete the oldest SnapInfo directory Snapshot copies).
-K	If the operation is a backup of the transaction log only (that is, if the **-L** option is specified), use the following option if you want to delete the oldest SnapInfo directory Snapshot copies: **-K** *NumberOfSnapInfoSnapshotsToKeep* *NumberOfSnapInfoSnapshotsToKeep* is the number of SnapInfo directory Snapshot copies to keep. This option is not valid unless you also specify the **-J** option (create a Snapshot copy of the SnapInfo directory after the backup finishes).
-L	If you want to perform only a transaction log backup, use this option. **Note** If you want to perform a full database backup followed immediately by a transaction log backup, use the **-G** option instead.

Option	Description
-M	If you want to perform only verification of the most recent unverified backup Snapshot copies (no database or transaction log backup), use the following option: **-M** *NumberOfBackupSnapshotsToVerify* *NumberOfBackupSnapshotsToVerify* is the number of backup Snapshot copies you want to verify. If you want to perform a transaction log only backup followed by a backup verification, use the **-L** option (back up transaction log only) instead. To include backup verification *as part of a backup*, use the **-Daily**, **-Weekly**, or **-All** option; *do not* use the **-M** option. If you want to perform a full database backup (with or without a transaction log backup) followed by a backup verification, do not use either the **-L** option or the **-M** option. To include backup verification *as part of a backup*, use the **-Daily**, **-Weekly**, or **-All** option; *do not* use the **-M** option.
-N	If the operation includes database or transaction log verification, use this option if you *do not* want the Snapshot copies verified after the backups finish. **Note** When performing a backup from the command-line using **SmSqlBi**, the default behavior is to verify Snapshot copies after the backup finishes; the **-N** option can be used to *disable* this. Conversely, when performing a backup using the Backup Wizard or the Backup and Verification tab in the SnapManager graphical user interface, verification of backups is not performed unless the "Verify Snapshot after backup (run DBCC)" option is *enabled*.
-O	If the operation is a backup of the transaction log only (that is, if the **-L** option is specified), use this option if you want to remove inactive entries from the transaction logs.

Option	Description
-Q	If the operation includes database or transaction log verification, use this option if you want to specify that a different SQL Server be used to run the DBCC utility: **-Q** *VerificationServer* [**-I** *UserName* **-W** *Password*] *VerificationServer* is the host name of the verification server. *UserName* is the SQL Server account name (specified if the verification server is to be accessed using a different account from that used to access the production SQL Server). *Password* is the SQL Server account password (specified if the verification server is to be accessed using a different account from that used to access the production SQL Server).
-mpdir	Use specified mount point directory. Specify which mount point directory a Snapshot copy will be mounted to during database verification.
-mp	Mount Snapshot to a directory. Mount snapshots in a default NTFS directory during database verification. This option is also effective when there are no available drive letters to mount snapshots.
-R	If the operation is a full database backup in which you will be deleting the oldest full database backups (that is, for a full database backup that includes the option **-D** *NumberOfBackups*), use this option if you want to retain up-to-the-minute restore ability for older backups in other management groups.
-Recent	If the operation includes database backup, use this option if you want the database backup Snapshot copies and SnapInfo backup Snapshot copies to be named using the generic naming convention. If you do not use this option, unique naming convention (timestamped) is used. For more information, see "SnapManager Snapshot naming conventions" on page 142.

Option	Description
-RS	If you want to run a command after the operation finishes, use the following option: **-RS** *RCServer* **-RP** *CommandPath* **-RA** *ArgumentList* *RCServer* is the host name of the SQL Server on which the command is to be run. *CommandPath* is the full path to the program or script that you want to run. *ArgumentList* is the string of SnapManager operation-specific information to be passed to your program or script. For details, see "Command arguments pass information to your program or script" on page 341.
-UM	If the operation uses backups that reside on volumes configured as SnapMirror sources, use this option if you want to update the SnapMirror destination after the backup or verification operations finish.
-Weekly	If the operation includes database or transaction log verification, use this option if you want to specify that you only want to verify the backups assigned to the Weekly backup management group. Compare with the **-Daily** and **-All** options.
-X	If the operation includes database backup, use this option if you want to verify the live database *before* the backups are performed.
-Y	If the operation includes database backup, use this option if you want to verify the live database *after* the backups are performed.

Configuring SnapManager Application Settings C

This section describes SnapManager application settings that can be configured or changed at any time after SnapManager has been installed. The following topics are covered:

- "Overview of SnapManager application settings" on page 364
- "Connecting to an SQL Server instance" on page 366
- "Changing the SQL Server user account" on page 370
- "Database integrity verification options" on page 372
- "SnapManager backup options" on page 378
- "SnapManager restore options" on page 382
- "Run Command After Operation default settings" on page 385
- "Managing fractional space reservation" on page 387
- "Enabling or disabling database migration back to local disks" on page 398
- "SnapManager report directory options" on page 399
- "Event notification options" on page 401

Attention

You must run SnapManager from the system console, not from a Terminal Services client. Do not use Terminal Services for any type of SnapManager administration, because you might miss critical information that is displayed only in pop-up boxes at the system console.

This section *does not* describe the following topics:

- Settings that can be configured only with the Configuration Wizard
- Settings that are specific to the context of a particular backup or a database verification operation

This information is described in other, related topics in this guide.

Related topics:

- "Settings configurable only with the Configuration Wizard" on page 106
- Appendix A, "Tools for Managing Backup and Verification," on page 339

Overview of SnapManager application settings

The following table lists SnapManager application settings that can be configured or changed at any time after SnapManager has been installed. Shaded rows indicate settings that can also be configured using the Configuration Wizard.

Application setting	How the setting can be accessed
SQL Server to be managed[1]	◆ Action menu ◆ Tool bar button
SnapManager server identity	◆ Options menu ◆ Connect to SQL Server dialog box
Database verification settings[2]	◆ Options menu ◆ Tool bar button ◆ Backup Wizard ◆ Verification Wizard
Backup settings	◆ Options menu ◆ Tool bar button ◆ Backup Wizard ◆ Backup and Verification tab
Restore Settings	◆ Options menu ◆ Tool bar button ◆ Restore Wizard ◆ SnapManager Restore tab
Run Command After Operation[3]	◆ Options menu ◆ Within the context of a backup or database verification operation: ❖ Backup Wizard ❖ Backup and Verification tab
Migrate databases to local disk	Options menu
Report directory	Options menu
Event notification settings	Options menu

1. The first time you start SnapManager, the application automatically opens this dialog box to prompt you for this setting. See Chapter 4, "Starting SnapManager for the First Time," on page 67.
2. From the Configuration Wizard, you can specify only the verification server and security authentication method. In order to access other verification settings (the DBCC Options), you must open the Verification Settings dialog box.
3. When the Run Command After Operation dialog box is opened from the Actions menu, only the default settings can be viewed or configured. See "Run Command After Operation default settings" on page 385. However, from within the context of a specific operation, the default settings are presented and then can be modified for this operation only. As an option, the default settings can be updated. See "Using the Run Command After Operation dialog box" on page 345.

Connecting to an SQL Server instance

Use the Connect to SQL Server dialog box to connect to the SQL Server you want to manage and specify the security authentication method to be used to establish the connection.

- "About the Connect to SQL Server dialog box" on page 366
- "About SQL Server authentication" on page 366
- "Connecting to a different SQL Server" on page 367

About the Connect to SQL Server dialog box

Use the Connect to SQL Server dialog box to connect to a specific SQL Server. When you specify or change the settings in this dialog box, SnapManager immediately connects to the specified SQL Server using the specified security authentication method. These settings remain in effect as the default settings until or unless you change them.

When you start SnapManager, the Connect to SQL Server dialog box opens automatically if a *default SQL Server* has not yet been specified. You cannot proceed to use SnapManager until you successfully connect to an SQL Server. Thereafter, whenever the SnapManager application is started, it automatically connects to the default SQL Server using the default security authentication method.

If at a later time you want to manage a *different SQL Server*, explicitly open the Connect to SQL Server dialog box to connect to the remote SQL Server using an SQL Server account that belongs to the `Administrators` group on that machine.

About SQL Server authentication

The same SQL Server user name and password with sysadmin server role privileges should be present in all SQL Server instances.The following limitations apply to using SQL Server authentication.

While creating a scheduled backup or verification job: SnapManager prevents you from creating a scheduled backup or verification job using any of the following types of SQL Server Agent:

- An SQL Server Agent running as a cluster resource
- An SQL Server Agent running on a remote computer
- An SQL Server Agent running under the local system account

When using SQL Server authentication: When SQL Server authentication is used, SnapManager encrypts the SQL Server account password using the specified Windows account credentials.

- In the case of SQL Server Agent jobs, SnapManager fails to decrypt the SQL Server account password if the Windows account under which the SQL Server Agent service runs is changed.
- In the case of Windows Scheduled Task jobs, SnapManager fails to decrypt the SQL Server account password if the task "Run As" account is changed later.

Connecting to a different SQL Server

To connect to a different SQL Server, complete the following steps.

Note

This will change only the SnapManager server identity on the *current machine* and will not change the SnapManager server identity on the *remote host*.

Step	Action
1	Open the Connect to SQL Server dialog box. ◆ If the default SQL Server has not yet been specified, SnapManager automatically opens the dialog box when you start SnapManager. In this case, the SQL Server box is empty. ◆ If SnapManager is already connected to the default SQL Server, open the dialog box explicitly. To do this, go to the menu bar and select Action > Connect to Another SQL Server. In this case, the SQL Server box displays the name and connection authentication type of the currently connected SQL Server.
Select the SQL Server you want to manage.	
2	In the SQL Server box, specify the SQL Server computer instance you want to manage. ◆ If you want to connect to the single SQL Server instance on the local host, select `[local]`. ◆ Otherwise, select the name of the SQL Server or SQL Server instance you want to manage.

Step	Action
Select the security authentication method to be used.	
3	In the Connection panel, choose the security authentication method to be used to connect to the SQL Server. ◆ Windows authentication ◆ SQL Server authentication If you choose Windows authentication mode (the default selection), users with a valid Windows account can log in to Microsoft SQL Server without supplying a user name and password. Windows Authentication relies on the user being authenticated by the operating system and takes advantage of Windows user security and account mechanisms. **Note** Windows Authentication is the authentication mode recommended by Microsoft.
4	If you selected SQL Server authentication, you must also specify the login name and password. For more information, see "SnapManager service account requirements" on page 26.
Optional: Change the SQL Server user account to be used.	
5	If you want to connect to the SQL Server using a different user account, click Change to open the Server Identity dialog box. **Note** To access the Reports on another SQL Server, you must connect using an SQL Server user account that belongs to the `Administrators` group on that machine. For more information, see "SnapManager creates a share for remote access to the Report directory." on page 68. For details about using the Server Identity dialog box, see "Changing the SQL Server user account" on page 370.

Step	Action
Connect to the SQL Server.	
6	Click OK. **Result:** The Connect to SQL Server dialog box closes, and SnapManager connects to the specified SQL Server using the specified security authentication method. **Note** Whenever you connect to an SQL Server, SnapManager checks that the SQL Server computer is running the supported versions of SQL Server, Microsoft Data Access Components (MDAC), and SnapDrive. These checkpoints are described within the procedure described in "Starting SnapManager for the First Time" on page 67.
7	If, instead of connecting to the specified SQL Server, SnapManager displays an error message regarding the SQL Server, MDAC, or SnapDrive version on that SQL Server computer, do the following: **a.** Refer to "Windows host system requirements" on page 23 to determine which software components you need to update on the SQL Server computer. **b.** Close the SnapManager application. **c.** Upgrade the software on the SQL Server computer as needed. **d.** Restart SnapManager.

Changing the SQL Server user account

You can change the SnapManager service account used to run SnapManager backup and restore operations.

To change the SnapManager service account used to run SnapManager backup and restore operations, complete the following steps.

Step	Action
1	Open the Local SnapManager Server Properties dialog box. There are two ways to do this: ◆ From the SnapManager main window, go to the menu bar and select Options > Run SnapManager As. ◆ From the Connect to SQL Server dialog box, click Change. **Result:** The Local SnapManager Server Properties dialog box appears and displays the name of the current SnapManager account for this SQL Server instance.
2	Select the SnapManager service account that you want to use to run SnapManager on this system: ◆ If you want to change the SnapManager service account on this instance, select the The Launching User option. ◆ If you want to make SnapManager accessible from a remote SQL Server instance or within an MSCS environment, select the This User option. This option is selected by default.
3	If you select the This User option, also do the following: **a.** Either enter the user account in the User box in *DomainName\UserName* format or click Browse to select the domain and user name. **b.** In the Password and the Confirm Password boxes, enter the password for this user account.

Step	Action
4	Click OK. This change will not take effect until you exit this SnapManager session and any currently running SnapManager jobs have completed. After you start a new SnapManager session, any SnapManager operations, such as backup and restore, will run under this newly specified account.

Database integrity verification options

Use the Verification Settings dialog box to specify the verification server and configure database verification options. See the following topics for more information:

- "Selecting the database verification server" on page 372
- "Selecting DBCC options" on page 374
- "Using the Access LUN in Snapshot tab" on page 376

Note

When you change the database verification server, this change does not affect any database backup (with verification) or database verification only jobs that are already scheduled.

Selecting the database verification server

To view or change the verification server, complete the following steps *from the production SQL Server* and not from a remote verification server.

Step	Action
1	If you are specifying a remote verification server, be sure it is set up properly, as described in "Requirements for a remote verification server" on page 29.
2	From the *production SQL Server* (and not from a remote verification server), open the Verification Settings dialog box. ◆ From the SnapManager menu bar, select Options > Database Verification Settings. ◆ From the SnapManager tool bar, click the Verification Settings icon. ◆ From the Backup Wizard, go to the View or Change Verification Settings screen, and then click Verification Settings. ◆ From the Restore Wizard, go to the Verify the Databases in this Backup screen, and then click Verification Settings. **Result:** The Verification Settings dialog box appears. The SQL Server tab is active by default and displays the host name of the current verification server.

Step	Action
3	In the SQL Server box, specify the SQL Server instance you want to use as the database verification server. **Note** If you plan to specify a remote verification server, ensure that the server is set up properly, as described in "Requirements for a remote verification server" on page 29.
4	In the Connection panel, choose the security authentication method to be used to connect to the SQL Server. ◆ Windows authentication ◆ SQL Server authentication If you choose Windows authentication mode (the default selection), users with a valid Windows account can log in to Microsoft SQL Server without supplying a user name and password. Windows Authentication relies on the user being authenticated by the operating system and takes advantage of Windows user security and account mechanisms. **Note** Windows Authentication is the authentication mode recommended by Microsoft.
5	If you selected SQL Server authentication, also specify the login name and password. For more information, see "SnapManager service account requirements" on page 26.
6	Click OK. **Result:** The Verification Settings dialog box closes. **Note** Until you change these settings, database verification is run from the SQL Server you selected using the options you specified. It does not necessarily run on the system from which you opened the Verification Settings dialog box. It does not affect any database verification jobs that are already scheduled.

Selecting DBCC options

To specify which DBCC options are used to verify database backup Snapshot copies, complete the following steps.

Step	Action
1	Open the Verification Settings dialog box. ◆ From the SnapManager menu bar, select Options > Database Verification Settings. ◆ From the SnapManager tool bar, click the Verification Settings icon. ◆ From the Backup Wizard, go to the View or Change Verification Settings screen, and then click Verification Settings. ◆ From the Restore Wizard, go to the Verify the Databases in this Backup screen, and then click Verification Settings. **Result:** The Verification Settings dialog box appears. The SQL Server tab is active by default and displays the host name of the current verification server.
2	Click the DBCC Options tab. **Result:** The DBCC Options tab displays the currently selected DBCC options.
3	In the DBCC Options panel, select all of the options you want to use: ◆ `NOINDEX` ◆ `ALL_ERRORMSGS` ◆ `NO_INFOMSGS` ◆ `TABLOCK` ◆ `PHYSICAL_ONLY` For more information about these options, see your Microsoft SQL Server documentation. **Note** By default, none of these options is selected.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">4</td><td colspan="2">By default, the database is detached after the DBCC utility finishes.</td></tr>
<tr><td>If...</td><td>Then...</td></tr>
<tr><td>If you want to detach the database after the verification finishes</td><td>Deselect the Leave Database Attached After Verification option.</td></tr>
<tr><td>If you want to leave the database attached after the verification finishes</td><td>Select the Leave Database Attached After Verification option.

If a database verification (with or without a full database backup) is started or scheduled with this option enabled, a message box will notify you that this option is set and will prompt you to confirm that you want to continue. Unless you explicitly detach the database and dismount the Snapshot copy LUNs after this operation completes, subsequent backup operations on this database will encounter busy Snapshot copies.</td></tr>
<tr><td>5</td><td colspan="2">Click OK.

Result: The Verification Settings dialog box closes.</td></tr>
</table>

Using the Access LUN in Snapshot tab

Use the Access LUN in Snapshot tab to specify how SnapManager is to access the database backup Snapshot copies during database integrity verification.

Related topics:

- "Selecting the database verification server" on page 372

To specify which method to use to access database backup Snapshot copies during the database integration verification, complete the following steps.

Step	Action	
1	Click the Access LUN in Snapshot tab.	
	From the...	**Do this...**
	Configuration Wizard	Go to the Database Verification and Account Authentication screen and select Advanced.
	SnapManager Main menu bar	Select Options > Database Verification Setting, and then click the Access LUN in Snapshot tab.
	Backup Wizard	Go to the Verify the Databases and Transaction Logs in this Backup screen, click Verification Settings, and then click the Access LUN in Snapshot tab.
	Restore Wizard	Go to the Verify Databases in this Backup screen, click Verification Settings, and then click the Access LUN in Snapshot tab.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td colspan="2">Assign either a drive letter or a directory path to access the backup Snapshot copy as a mounted LUN. By default, the default mount directory path appears as follows:

“C:\ProgramFiles\NetApp\SnapMaanger for SQL Server\SnapMgrMountPoint”.</td></tr>
<tr><td></td><td>If you want to...</td><td>Then...</td></tr>
<tr><td></td><td>Mount the Snapshot copy on the next available drive letter</td><td>Select the “Automatically assign available drive letter” option.</td></tr>
<tr><td></td><td>Mount the Snapshot copy on a specific NTFS mount point</td><td>Do the following:

a. Select the “Mount in an empty NTFS directory” option.

b. Enter or browse to the directory path of an NTFS mount point.

Note
This mount point will be used, if SnapManager is configured to use drive letters but runs out of available drive letters.</td></tr>
<tr><td>3</td><td colspan="2">Click OK.

Result: The Verification Settings dialog box closes. After the database verification, the Snapshot copy directory created in the default mount point directory path is dismounted automatically.</td></tr>
</table>

SnapManager backup options

Use the Backup Settings dialog box to configure default settings for SnapManager backup operations. See the following topics for more information:

- "Configuring the profile of a full database backup" on page 378
- "Configuring the profile of a transaction log backup" on page 380

Configuring the profile of a full database backup

To configure the profile of a full database backup, complete the following steps.

Note

For a complete list of parameters that are applied to a full database backup operation, see "Information you need to specify for a full database backup" on page 152.

Step	Action
1	Open the Backup Settings dialog box. ◆ From the SnapManager menu bar, select Options > Backup Settings. ◆ From the SnapManager tool bar, click the Backup Settings icon. ◆ From the Backup Wizard, go to the Other Options screen and then click Backup Settings. **Result:** The Backup Settings dialog box appears. The Full Database Backup tab is active by default and displays the current settings.
2	In the Maximum Databases to Back Up Concurrently box, specify the maximum number of databases that can be backed up concurrently. The default value is 255. However, this value can also be configured in the Configuration Wizard.

Step	Action
3	In the "Select a backup naming convention..." panel, specify the naming convention you want used to form database backup Snapshot copy names and SnapInfo directory Snapshot copy names. ◆ If you want the most recent backup to be identified by the Snapshot copy name that includes the string `recent`, select the "Use Generic..." option. This option is selected by default. ◆ If you want all Snapshot copy names—even for the most recent backup—to contain the Snapshot copy creation date and time, select the Use Unique option. This is the Snapshot copy naming convention used by SnapManager 1.0 for Microsoft SQL Server. For more detailed information, see "SnapManager Snapshot naming conventions" on page 142.
4	In the "Select run verification with live database" panel, select whether you want to run verification against the live database before the backup, after the backup, or both before and after the backup. **a.** If you want to run the DBCC utility against the live database before the database is backed up, select the Run DBCC Physical Integrity Verification Before the Backup option. **b.** If you want to run the DBCC utility against the live database after the database is backed up, select the Truncate Committed Transactions in the Transaction Log option. **Note** By default, both options are not selected because database verification is a time-consuming activity.
5	To apply your changes and close the dialog box, click OK. **Result:** The new settings will be applied to all subsequent full database backups.

Configuring the profile of a transaction log backup

To configure the profile of a transaction log backup, complete the following steps.

Note

For a complete list of parameters that are applied to a transaction log backup operation, see "Information you need to specify for a transaction log backup" on page 172.

Step	Action
1	Open the Backup Settings dialog box. ◆ From the SnapManager menu bar, select Options > Backup Settings. ◆ From the SnapManager tool bar, click the Backup Settings icon. ◆ From the Backup Wizard, go to the Other Options screen and then click Backup Settings. **Result:** The Backup Settings dialog box appears. The Full Database Backup tab is active by default.
2	In the dialog box, click the Transaction Log Backup tab. **Result:** The Transaction Log Backup tab displays the current settings.
3	If you want to create a Snapshot copy of the SnapInfo subdirectory for this backup after the backup finishes, select the "Create Snapshot..." option.
4	If you create a Snapshot copy of the SnapInfo subdirectory for this backup after the backup finishes, you also have the option to automatically delete the oldest Snapshot copies of the SnapInfo subdirectory for this backup. To choose this, do the following: **a.** Select the "Delete Oldest Snapshots..." option. **b.** Select the number of Snapshot copies you want to retain.
5	If you want to remove inactive entries from the transaction log, select the "Remove Inactive Entries..." option.

Step	Action
6	To apply your changes and close the dialog box, click OK. **Result:** The new settings will be applied to all subsequent full database backups.

SnapManager restore options

Use the Restore Settings dialog box to configure default settings for SnapManager restore operations. See the following topics for more information:

- "Understanding the restore options" on page 382
- "Configuring the profile of a restore operation" on page 383

Understanding the restore options

The following table describes each of the database restore options.

Restore option	Description	Default
Recover database without restoring at the end of restore, if needed	If this option is selected and the database is not fully operational, SnapManager skips the restore and instead recovers the database to its previous state.	Not selected
Restore databases even if existing databases are online	If this option is selected and an existing database is online at the time of the restore operation, SnapManager proceeds with the restore and overwrites the existing database.	Not selected
Retain SQL database replication settings	If this option is selected and you are restoring databases for an SQL Server instance that is acting as a Publisher or as a Subscriber in a replication topology, the replication relationship is retained after the SnapManager restore operation finishes.	Not selected
Create transaction log backup before restore	If this option is not selected, SnapManager does not create a transaction log backup before the restore is performed, thereby decreasing overall restore time. Deselect this option under the following circumstances: ◆ You already have another restore Snapshot copy of the transaction log and you want to decrease restore time. ◆ You are recovering from a mirrored backup for which the transaction log files were lost. Disabling this option avoids subsequent creation of SnapManager backup sets on a recovery path that is inconsistent with that of the database. ◆ You are restoring a log-shipped database.	Selected

Restore option	Description	Default
Abort database restore if transaction log backup before restore fails	This option is available if the "Create transaction log backup before restore" option is selected. If this option is selected and the transaction log backup before restore fails, SnapManager aborts the database restore operations.	Not selected

Configuring the profile of a restore operation

To configure the profile of a restore operation, complete the following steps.

Note

For a complete list of parameters that are applied to a database restore operation, see "Information you need to perform a restore" on page 226.

Step	Action
1	Open the Restore Settings dialog box. ◆ From the SnapManager menu bar, select Options > Restore Settings. ◆ From the SnapManager tool bar, click the Restore Settings icon. ◆ From the Restore Wizard, go to the Other Options screen and then click Restore Settings. **Result:** The Restore Settings dialog box appears. The Restore Options tab is active by default.
2	Select any combination of the restore options you want to use: ◆ Recover database without restoring at the end of restore if needed ◆ Restore databases even if existing databases are online ◆ Retain SQL database replication settings ◆ Create transaction log backup before restore ◆ Abort database restore if transaction log backup before restore fails These options are described in "Understanding the restore options" on page 382.

Step	Action
3	Click OK to apply your changes and close the dialog box. **Result:** The new settings will be applied to all subsequent database restore operations.

Run Command After Operation default settings

You can configure default values that you want used to prepopulate the Run Command After Operation dialog box when it is opened from either the Backup and Verification tab or the Backup Wizard.

Related topics:

- "Archiving SnapManager backups using a Windows backup utility" on page 274
- "Automatically archiving SnapManager backups using SnapVault" on page 278
- "Using the Run Command After Operation dialog box" on page 345

To use the Run Command After Operation dialog box to configure default settings, complete the following steps.

Step	Action
1	Open the Run Command After Operation dialog box from the Option menu. To do this, select Options > Default Run Command After Operation.
2	In the "Specify a computer where..." box, enter or browse to the name of the host on which your program or script resides.
3	In the "Specify the full path..." box, browse to your program or script.

Step	Action
4	Form the command input string in the Command Arguments box. You can do this using any combination of the following methods: ◆ To enter text directly into the Command Arguments box, click in the box and type the desired text. ◆ To enter a SnapManager variable into the Command Arguments box, do the following: **a.** If necessary, click in the Command Arguments box to position the cursor. **b.** In the SnapManager Variables list, select the variable you want to enter. For more information, see "Command arguments pass information to your program or script" on page 341 and "Command arguments are operation specific" on page 342. **c.** Click Select. **Note** Both the `$SnapInfoPath` and `$LogBackupFile` variables are automatically enclosed within double quotes so that the actual path name can contain spaces without affecting the script invocation on the Windows command-line. If you do not want the double quotes to appear in your command-line, remove them from the Command Arguments field in the Run Command After Operation dialog box.
5	Repeat Step 4 as needed until the Command Arguments box contains the arguments you want to pass to your program or script.
6	Click OK to apply your changes and close the Run Command After Operation dialog box. **Result:** Whenever the Run Command After Operation dialog box is opened from either the Backup and Verification tab or the Backup Wizard, the boxes will be prepopulated with the values you specified as default settings.

Managing fractional space reservation

The following sections explain Data ONTAP space reservation and describe how to configure SnapManager for SQL fractional space reservation policies:

- "Understanding fractional space reservation" on page 387
- "Fractional space reservation policy settings in SnapManager for SQL" on page 388
- "About the default fractional space reservation policy" on page 390
- "Viewing fractional space reservation status" on page 391
- "Configuring fractional space reservation policies" on page 394

Understanding fractional space reservation

This section summarizes space reservation and fractional space reservation as supported by Data ONTAP 7.1 or higher. For more detailed information about these features, see the *Data ONTAP Block Access Management Guide for iSCSI and FCP* for release 7.1 or higher.

Space reservation: When you create a LUN on a storage system volume, Data ONTAP reserves enough space in the traditional or flexible volume so that write operations to those LUNs do not fail due to lack of disk space on the storage system. Other operations, such as making a Snapshot copy or creating new LUNs, can occur only if there is enough available unreserved space. These operations are restricted from using reserved space.

SnapDrive creates and manages LUNs with space reservation enabled. That is, additional space on the storage system volume is automatically reserved for overwriting blocks that belong to a LUN. By default, this additional space is equal to 100 percent of the total size of all space-reserved LUNs in the storage system volume.

If space reservation is disabled, write operations to a LUN might fail due to insufficient disk space in the storage system volume; consequently, the host application might terminate, report I/O errors, or exhibit unexpected behavior.

Fractional space reservation: With fractional space reservation, the amount of space reserved for overwrites is set to less than 100 percent of the total size of all space-reserved LUNs in a traditional volume or a flexible volume that has the guarantee option set to `volume` rather than `file`. The space that is preallocated for space reservation is reduced to that percentage. Fractional space reservation is generally used for volumes with LUNs that store data with a *low rate of change*.

While space reservation is enabled at the LUN level, fractional space reservation overwrite reserve amounts are configured at the volume level. Fractional space reservation does not control how the total amount of space reserved for overwrites in a volume is applied to individual LUNs in that volume.

Fractional space reservation policy settings in SnapManager for SQL

In a SnapManager environment in which SQL data is stored on LUNs in a fractional space-reserved storage system volume, you need to avoid an out-of-space condition. You can do this using the SnapManager space reservation policy tool to monitor the overwrite reserve utilization on the volumes. Space reservation has two policies that you can enable:

- Automatic deletion of SQL backups
- Automatic dismounting of SQL databases

These policies control whether SnapManager takes action to prevent the overwrite reserve from being fully depleted when the overwrite reserve space runs low on a fractional space reserved volume.

Note

If you enable SnapManager e-mail notification, SnapManager sends an SMTP e-mail message to specified recipients after a SnapManager fractional space reservation policy event completes.

Automatic deletion of SQL backups: SnapManager provides an option for automatic deletion of backups of LUNs that store SQL data. When you enable this option, the SnapManager fractional reservation policy serves as the SQL-aware replacement for or adjunct to the Data ONTAP Snapshot copy deletion feature. If the level of overwrite reserve utilization on the volume reaches a threshold specified by the policy, automatic backup deletion is triggered and SnapManager deletes SQL backups using the following guidelines:

- Delete the oldest backup Snapshot copies first.
- Retain the specified number of total backup Snapshot copies on the volume.
- Retain the most recent backup of any database (if it resides on the volume).
- Retain backup of any databases that are no longer in existence.
- Retain up-to-the-minute restore ability for the older backups in other backup management groups. This option can be enabled only if “Delete Oldest backup Snapshot first” is selected.

Select the backup retention level that is appropriate to your SnapManager backup creation and verification schedule. If Snapshot copy deletion triggers, enough backup Snapshot copies should be retained so that *at least one verified backup*

remains on the volume. Due to these SQL-aware features, the automatic deletion of Snapshot copies does not necessarily prevent an out-of-space condition on the volume.

Automatic dismounting of SQL databases: When this option is enabled, SnapManager provides an option for the automatic dismounting of SQL databases in space-reserved LUNs. It is triggered if overwrite reserve utilization on the volume reaches the threshold specified by the fractional space reservation policy. This stops SQL write operations to LUNs in a storage system volume for which overwrite reserve space is nearly full.

This second component of the fractional space reservation policy should be always enabled. It is a last-resort action that prevents further consumption of overwrite reserve.

When both options of fractional space reservation policy are enabled, the *dismounting of SQL databases* must be triggered at a *higher level of overwrite reserve utilization* than is used to trigger the *deletion of SQL backup Snapshot copies*. This enables SnapManager to first use backup-set deletion to free some overwrite reserve. If this is not sufficient, dismounting the affected database prevents further consumption of overwrite reserve.

Attention

If another host or client continues to write data to the affected volume, the overwrite reserve space might still run out and the storage system volume will go offline. For this reason, it is recommended that dedicated volumes be used for SQL data.

Fractional space reservation policy settings: The following table summarizes the fractional space reservation policy by listing each setting, along with its factory default value and its configurable values.

SnapManager fractional space reservation policy setting	Factory default value	Configurable values
Deletion of SQL backup Snapshot copies		
Status	Enabled	Enabled or disabled [1]
Trigger on overwrite reserve utilization	70%	1% – 99% [2]
Number of Snapshot copies to retain	5	1 – 256
Dismounting of SQL databases		
Status	Always enabled [1]	
Trigger on overwrite reserve utilization	90%	1% – 99% [2]

[1] Enabling automatic deletion of SQL backup Snapshot copies does not necessarily prevent an out-of-space condition on the volume. Therefore, database dismounting is always enabled.

[2] Enabling automatic deletion of SQL backup Snapshot copies does not necessarily prevent an out-of-space condition on the volume. Therefore, if Snapshot copy deletion is enabled, it must be configured to trigger *prior to* database dismounting.

About the default fractional space reservation policy

The default fractional space reservation policy is automatically enabled for any traditional or flexible storage system volume that has overwrite reservation set to less than 100 percent and contains LUNs that store SQL database files, SQL transaction log files, or SnapManager SnapInfo directories.

Default policy with default settings: You can use the default policy as-is, allowing the factory default values to be applied to every storage system volume that contains fractional-space-reserved LUNs.

Default policy with customized settings: You can customize the default policy that is applied to all storage system volumes that contain fractional-space-reserved LUNs. You can configure the default policy settings to required values. This will be applied to the storage system volume if the “Apply default policy to selected object in the left window” check box is selected in the Fractional Space Reservation Settings screen.

Volume-specific policies: You can override the default policy for any particular volume that contains fractional-space-reserved LUNs by applying a custom policy.

For more information on how to configure volume-specific policies, see "Configuring fractional space reservation policies" on page 394

Viewing fractional space reservation status

In the Fractional Space Reservation Settings dialog box, use the Current Status tab to view the current space consumption in the storage system volumes that contain LUNs that store SQL data or SnapInfo directories. You will see the following information:

Drive Letter or Mount Point: *A SnapManager configuration setting for the LUN.* The drive letter or NTFS mount point on which the LUN is mounted.

Fractional Overwrite Reserve (%): The amount of space reserved for overwrites on the storage system volume that contains the LUN. It is expressed as a percentage of the total size of all space-reserved LUNs in the volume.

Backup AutoDelete Trigger (%): *A SnapManager fractional space reservation policy setting for the storage system volume that contains the LUN.* It is expressed as a percentage of overwrite reserve utilization that triggers automatic deletion of SQL backup sets.

Disable Database Trigger (%): *A SnapManager fractional space reservation policy setting for the storage system volume that contains the LUN.* It is expressed as a percentage of overwrite reserve utilization that triggers automatic disabling of SQL databases.

Used Overwrite Reserve (%): The amount of overwrite reserve *in use* for the storage system volume that contains the LUN. It can be expressed in two ways:

- As a percentage of the total size of all space-reserved LUNs in the volume
- In megabytes representing the total size of all space-reserved LUNs in the volume

Available Reserve (MB): This is the amount of overwrite reserve available for the storage system volume that contains the LUN.

Storage System Snapshot Autodelete: This state of the Data ONTAP Snapshot copy autodeletion feature: enabled or disabled for the storage system volume that contains the LUN. If this LUN stores SQL data files and is contained in a storage system volume for which the Data ONTAP Snapshot copy autodeletion feature is enabled, disable this feature on that volume or ensure that it is configured so that it will not delete SnapManager backup set components.

To view the current space consumption information about each LUN, complete the following steps.

Step	Action
1	In the SnapManager main window, select Options > Fractional Space Reservation Settings.
2	In the Current Status tab, note the space consumption status for each LUN that stores SQL data or SnapInfo directories. ◆ The following columns displays SnapManager configuration information: ❖ Drive Letter or Mount Point ❖ Backup Autodelete Trigger (%) ❖ Disable Database Trigger (%) **Note** The SnapManager fractional space reservation policy triggers (listed above) are not applicable to fully space-reserved LUNs. ◆ The following columns display the fractional overwrite reserve settings and status: ❖ Fractional Overwrite Reserve (%) ❖ Used Overwrite Reserve (%) ❖ Used Overwrite Reserve (MB) ❖ Available Reserve (MB) ❖ Storage System Snapshot Autodelete **Note** If Fractional Overwrite Reserve (%) is 100, the LUN is contained in a fully space-reserved volume rather than a fractionally space-reserved volume. The information displayed in this tab automatically refreshes every 60 seconds. **Note** Only the Drive Letter or Mount Point column displays LUN-specific information. All other columns in this tab display information that applies across the storage system volume that contains the LUN.

Step	Action
3	If the Snapshot Autodelete column is `Enabled`, investigate the cause and take preventive action if necessary. **Attention** If the Storage Snapshot Autodelete column is `Enabled`, the LUN is contained in a FlexVol™ volume that has overwrite reserve set to less than 100 percent and that also has the Data ONTAP automatic Snapshot copy deletion feature enabled and configured to trigger when the overwrite reserve is nearly full. If SQL data or SnapManager SnapInfo directories are stored on LUNs contained in a volume with these features, the Data ONTAP Snapshot copy autodeletion policy might delete SQL backup set components. If the Storage Snapshot Autodelete column is `Enabled`, take one of the following actions on the volume: ◆ Disable the Data ONTAP Snapshot copy autodelete feature. ◆ Ensure that the Data ONTAP Snapshot copy autodelete feature is configured in such a way that it will not delete SQL backup set components. For details about the `snap autodelete` storage system command, see the *Data ONTAP® Block Access Management Guide for iSCSI and FCP* for release 7.1 or later. **Note** The SnapManager fractional space reservation policy includes a separate, SQL-aware autodeletion feature. For details, see "Automatic deletion of SQL backups" on page 388 and "Configuring fractional space reservation policies" on page 394. The SnapManager autodeletion feature can be used in place of or in conjunction with the Data ONTAP autodeletion feature; you can also choose to disable the SnapManager autodeletion feature.
4	To close the dialog box, click OK.

Configuring fractional space reservation policies

In the Fractional Space Reservation Settings dialog box, use the Policy Settings tab to view or customize the default policy and to configure custom policies for individual fractional-space-reserved LUNs.

The default fractional space reservation policy and its factory default settings are described in Appendix C, "About the default fractional space reservation policy," on page 390.

Note

SnapManager automatically applies the default policy to every storage system volume that contains fractional-space-reserved LUNs that store SQL database files or SnapInfo directories. Therefore, your storage is protected from an out-of-space condition, without requiring you to explicitly enable or configure any fractional space reservation policies.

To configure the fractional space reservation policy, complete the following steps.

Step	Action	
1	In the SnapManager main window, select Options > Fractional Space Reservation Settings.	
2	Select the Policy Settings tab.	
Choose to specify either the default policy or a volume-specific policy		
3	In the left navigation tree, select the scope of the policy you want to view or change in the main panel on the right-hand side of the tab:	
	If you want to view or change...	**Then do this...**
	The default policy	In the navigation tree, select Default Policy.
	A volume-specific policy	In the navigation tree, select the storage system and then the volume.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Disable or configure automatic deletion of SQL backup Snapshot copies</td></tr>
<tr><td>4</td><td>Use the “Automatic deletion of backups” panel to disable, enable, or configure automatic deletion of SQL backup Snapshot copies in fractional-space-reserved LUNs on the volume.

Note
Although automatic deletion of SQL backup Snapshot copies does not necessarily prevent an out-of-space condition on the volume, it is recommended that this feature be enabled for every volume that contains fractional-space-reserved LUNs that store SQL data.

For more information, see “Automatic deletion of SQL backups” on page 388.

<table>
<tr><th>If you want to...</th><th>Then do this...</th></tr>
<tr><td>Enable automatic deletion of SQL backup Snapshot copies</td><td>Select the “Delete backups that include LUNs which have less than 100% overwrite reservation” option, and then skip to Step 7.</td></tr>
<tr><td>Disable automatic deletion of SQL backup Snapshot copies</td><td>De-select the “Delete backups that include LUNs which have less than 100% overwrite reservation” option, and then proceed to Step 5.</td></tr>
</table>
Note
Data ONTAP includes a separate Snapshot copy autodeletion feature. For details, see “Viewing fractional space reservation status” on page 391. The SnapManager autodeletion feature can be used in place of or in conjunction with the Data ONTAP autodeletion feature.</td></tr>
</table>

Step	Action
5	In the “Trigger point for overwrite reserve utilization” field, enter the level of overwrite reserve utilization (in percentage of total reserve) that is to trigger deletion of SQL backup Snapshot copies. The value must be a non-negative integer that is less than the “Trigger point for overwrite reserve utilization” value in the “Automatically dismount databases” panel.
6	In the “Number of most recent backup sets to retain” field, enter the number of backup set to be retained, if automatic backup set deletion is triggered. The value must be an integer from 1 through 256 and should be based on the backup creation and verification schedule. For more information, see “Automatic deletion of SQL backups” on page 388”.

Step	Action
Configure automatic dismounting of SQL databases	
7	Use the “Automatically dismount databases” panel to configure automatic dismounting of SQL databases in fractional-space-reserved LUNs on the volume. **Note** The automatic deletion of SQL backup Snapshot copies does not necessarily prevent an out-of-space condition on the volume. SnapManager does not allow you to disable dismounting of databases for any fractional space reservation policy. In the “Trigger point for overwrite reserve utilization” field, enter the level of overwrite reserve utilization (in percentage of total reserve) that is to trigger dismounting of SQL databases. The value must be an integer from 0 through 99. **Note** If Snapshot copy autodeletion is enabled, SnapManager requires that this threshold be set to a higher level than the threshold that triggers automatic Snapshot copy deletion. This ensures that Snapshot copy autodeletion is triggered first. For more information, see “Automatic dismounting of SQL databases” on page 389.
Apply the changes to the default or volume-specific policy	
8	To apply your changes and close the dialog box, click OK.

Enabling or disabling database migration back to local disks

The primary function of the Configuration Wizard is to migrate SQL Server databases to Data ONTAP LUNs so that the databases can be backed up and restored using SnapManager. If you choose to stop using SnapManager as your data management tool, you can also use the Configuration Wizard to migrate your databases back to local disks. However, by default, the Where to Move databases screen of the Configuration Wizard does not list any local drives unless you explicitly enable the "migrate databases to local disks" feature.

Related topics:

- "Migrating SQL Server databases to LUNs" on page 107
- "Migrating SQL Server databases back to local disks" on page 126

To enable or disable the ability to migrate databases from Data ONTAP LUNs back to local disks, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">View the current state of the "migrate databases to local disks" feature from the Option menu. To do this, select Options > Configuration Wizard Option.

Result: The "Migrate Database(s) to Local Disk" option is listed.

If the feature is enabled, a check mark appears to the left of the menu option.</td></tr>
<tr><td rowspan="3">2</td><td colspan="2">To enable or disable the "migrate databases to local disks" feature, select Options > Configuration Wizard Option > Migrate Database(s) to Local Disk.</td></tr>
<tr><td>If...</td><td>Then...</td></tr>
<tr><td>The option was previously disabled</td><td>A dialog box appears and informs you that the migrate back to local disk feature is now enabled.</td></tr>
<tr><td>The option was previously enabled</td><td>A dialog box appears and informs you that the migrate back to local disk feature is now disabled.</td></tr>
<tr><td>3</td><td colspan="2">Click OK to close the dialog box.</td></tr>
</table>

SnapManager report directory options

Use the SnapManager Report Directory Setting dialog box to configure the location of the SnapManager report files. For more information, see any of the following topics.

- "Default report directory" on page 399
- "Reasons to change the report directory" on page 399
- "Accessing reports created in a previous directory" on page 400
- "Using the Report Directory Setting dialog box" on page 400

Related topics:

- "SnapManager creates a share for remote access to the Report directory." on page 68
- Chapter 11, "Managing SnapManager Operational Reports," on page 249

Default report directory

By default, the SnapManager reports are stored in a subdirectory named `Report` under the directory in which the SnapManager application is installed. If you installed SnapManager in its default installation directory, then the default report directory path is as follows:

```
C:\Program Files\NetApp\SnapManager for SQL Server\Report
```

Reasons to change the report directory

Reasons for changing the location of the SnapManager report directory are described in the following paragraphs.

Limited space: If you find you have limited space in the current report directory, you can change the report directory to a different location that has more available disk space.

Clustered environment: If you are running SQL Server and SnapManager in an MSCS cluster, storing the SnapManager reports in the default location (in a directory named `Report` under the SnapManager installation directory) would not allow the report directory to be shared between the nodes in the cluster. Furthermore, you would not see the same reports from different nodes.

To avoid these problems, you can move the report directory to a disk that belongs to the same group as your SQL Server virtual server. This needs to be performed from every SnapManager node.

Accessing reports created in a previous directory

If you change the name or location of the SnapManager report directory, you cannot use the SnapManager Reports tab to view or print any reports that were created in that report directory.

However, assuming the previous report directory was not explicitly changed or removed, any reports created in that directory are still accessible. In order to view or print those older reports, you must change the report directory back to its previous location.

Using the Report Directory Setting dialog box

Use the Report Directory Setting dialog box to view and change the directory where your SnapManager reports are stored.

Step	Action	
1	From the SnapManager menu bar, select Options > Report Directory Setting. **Result:** The Report Directory dialog box appears and displays the current location of the report directory.	
2	Specify the new location for the report directory. **Attention** Do not use a disk that contains SQL Server or SnapManager data for the report directory; it is restored from the Snapshot copy when you perform a SnapManager Restore.	
	If...	**Then...**
	You know the full directory path name	Click in the Report Directory box and modify the path name.
	You prefer to browse to the new location	Click Browse to use a browse dialog box to select the new location.
3	To apply your changes and close the Report Directory Setting dialog box, click OK.	
4	To refresh the information displayed in the SnapManager Reports tab, go the SnapManager menu bar and select View > Refresh.	

Event notification options

You can use either the Configuration Wizard or the Auto Notification Settings dialog box to enable and configure the SnapManager event notification services. See the following topics for more information:

- "Understanding SnapManager event notification options" on page 401
- "Using the Auto Notification Settings dialog box" on page 402

Understanding SnapManager event notification options

The following event notification options can be configured from either the Configuration Wizard or from the Auto Notification Settings dialog box.

SnapManager e-mail notification: SnapManager can notify you through e-mail (using SMTP) about the success or failure of the following types of events:

- SnapManager backup
- Database integrity verification
- SnapManager restore
- SnapManager configuration

SnapManager event logging: If AutoSupport is enabled on the storage system, the SnapManager events can be posted to the storage system event log. This option is enabled by default.

AutoSupport notification: If AutoSupport is enabled on both the storage system and SnapManager, technical support receives automatic e-mail notification about any SnapManager events or storage system problems that might occur. This option is enabled by default.

The AutoSupport daemon monitors the storage system's operations and sends automatic messages to technical support to alert them to potential storage system problems. If necessary, technical support contacts you at the e-mail address that you specified to help resolve a potential system problem. The AutoSupport daemon is enabled by default on the storage system. For additional information, see the *Block Access Management Guide* for your version of Data ONTAP.

Limit event logging to failure events: If AutoSupport is enabled on the storage system, you can limit the SnapManager events that are posted to the storage system event log and AutoSupport (if enabled for SnapManager) to failure events only. The option to limit event logging to failure events is enabled by default.

Using the Auto Notification Settings dialog box

To configure automatic event notification settings for SnapManager, complete the following steps.

Step	Action
1	From the SnapManager menu bar, select Options > Notification Settings. **Result:** The Auto Notification Settings dialog box appears. **Note** The Configuration Wizard presents the same options in the Configure Automatic Event Notification screen.
Configure E-mail Notification	
2	Optional. To enable e-mail notification, select the Send Email Notification option. By default, the automatic e-mail notification feature is disabled. **Note** SnapManager relies on and requires an external mail host at your site to send mail. The mail host is a host that runs a mail server that listens on the SMTP port (25).

Step	Action
3	In the four text boxes in the top half of the tab, enter the following information. **SMTP Server:** The host name or the IP address of the SMTP e-mail server or gateway to be used. **From:** The e-mail address of the sender of the notification. By default, the name `SMSQLAutoSender` is used. To specify a sender other than the default, use one of the following formats: ❖ *SenderAlias*<*SenderName@SenderDomain*.com> ❖ *SenderAlias* ❖ *SenderName@SenderDomain*.com **To:** The e-mail address of the recipient to whom the notification is to be sent. For more than one recipient, use a semicolon (;) to separate the addresses. **Subject:** The text to be appended to the following standard subject line, which is included in all notification messages: `Backup status at` *mm_dd_yyyy-hh.mm.ss* `from` *MachineName* By default, `SnapManager for SQL Server` is used for the appended subject string.
4	Click Advanced. **Result:** The Advanced Event Notification Settings dialog box appears.
Configure Advanced E-mail Notification Settings	
5	In the Operation Type panel, select all of the SnapManager operation types for which you want e-mail notifications sent: ◆ Backup & Verification ◆ Restore ◆ Configuration

Step	Action
6	In the E-mail Message Content panel, select one of the following types of body messages to include in the body of the e-mail: ◆ Send operation results summary **Note** If you choose to send the operational results in summary format rather than in verbose format, you can also select the Include SnapManager Operation Report as an Attachment option. ◆ Send verbose operation results
7	Click Apply to commit your settings.
8	Click Advanced to open the Advanced E-mail Notifications Settings dialog box. To configure the SMTP mail notification settings, select either or both of the following two options: ◆ Only send notification when operation fails Specifies that you want e-mail notification sent only when a backup process or a verification process fails (off by default). ◆ Include SnapManager operation report as attachment Specifies that you want the status report to be attached to the e-mail notification (off by default).
9	Click OK to apply your settings and close the Advanced Email Notification Settings dialog box.
10	Click Send a Test Email. **Result:** SnapManager sends the e-mail notification, using the settings you specified, and displays a notification.
Configure Event Logging and AutoSupport	
11	If you want to enable posting of SnapManager events to the storage system event log, select the Log SnapManager Events to Filer Syslog option.
12	If SnapManager event logging is enabled, you can also enable automatic e-mail notification about any SnapManager or storage system problems to technical support. To do this, select the Send AutoSupport Notification option.

Step	Action
13	If you want to limit SnapManager event logging to failure events, select the “On failure only” option.
14	Click OK. **Note** If you are using the Configuration Wizard instead of the Auto Notification Settings dialog box, click Next.

Configuration Wizard Screens and Dialog Boxes D

This section describes the Configuration Wizard screens and dialog boxes. The following topics are covered:

- "Using the Database Verification Server and Account Authentication screen" on page 408
- "Using the Where to Move Databases screen" on page 410
- "Using the Number of Databases that Can Be Backed Up Concurrently screen" on page 417
- "Using the Modify SQL Server Maximum Worker Threads Parameter screen" on page 419
- "Using the Specify a Single SnapInfo Directory screen" on page 423
- "Using the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen" on page 425
- "Using the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen" on page 428
- "Using the Where to Put the SnapInfo Directory for Each Database screen" on page 431
- "Using the Database Migration Options screen" on page 434

Related topics:

- Chapter 6, "Preparing to Migrate SQL Server Databases," on page 85
- Chapter 7, "Using the SnapManager Configuration Wizard," on page 101

Using the Database Verification Server and Account Authentication screen

Use the Database Verification Server and Account Authentication screen to specify the SQL Server you want to use to perform database integrity verification and the security authentication method to be used to establish the connection to this server.

Related topics:

- "How SnapManager checks database integrity in backup sets" on page 135
- "Migrating SQL Server databases to LUNs" on page 107
- "Using the Database Migration Options screen" on page 434

To specify the SQL Server you want used to verify the SQL Server databases in your SnapManager backups, complete the following steps from within the Database Verification Server and Account Authentication screen of the Configuration Wizard.

Step	Action
1	If you want to defer specification of the database verification server (or if you do not plan to perform database verification at all), do the following: **a.** Select the "Select a verification server later using the Options menu" dialog box. **b.** Click Next. **c.** Go to Step 5. Later, if you want to specify a verification server outside of the Configuration Wizard, go to the menu bar and select Options > Database Verification Settings. For a detailed description of the Verification Settings dialog box, see Appendix C, "Configuring SnapManager Application Settings," on page 363.

Step	Action
2	Otherwise, to specify a verification server now, go to the Verification Server field and select the SQL Server instance you want to use to perform database integrity verification. **Note** It is recommend that, for optimal performance, you use a different SQL Server as the verification server. If this is a MSCS environment, you cannot select an SQL Virtual Server as the database verification server.
3	In the Connection panel, choose the security authentication method to be used to connect to the SQL Server. ◆ Windows authentication ◆ SQL Server authentication If you choose Windows authentication mode (the default selection), users with a valid Windows account can log in to Microsoft SQL Server without supplying a user name and password. Windows Authentication relies on the user being authenticated by the operating system and takes advantage of Windows user security and account mechanisms. **Note** Windows Authentication is the authentication mode recommended by Microsoft.
4	If you selected SQL Server authentication, you must also specify the login name and password. For more information, see "SnapManager service account requirements" on page 26.
5	Click Next. **Result:** The Configuration Wizard proceeds to the Where to Move Databases screen.

Using the Where to Move Databases screen

Use the Where to Move Databases screen to view the current configurations of the databases and to optionally specify how you want to move these databases. See the following topics for more information:

- "Viewing the current locations of SQL Server databases" on page 410
- "Migrating SQL Server databases from local disk to LUNs" on page 411
- "Moving SQL Server databases to different LUNs" on page 413
- "Viewing the full path for a database on a LUN" on page 416

Related topics:

- "SQL Server configuration rules with SnapManager" on page 86
- "SQL Server configurations supported with SnapManager" on page 89
- "Migrating SQL Server databases to LUNs" on page 107

Viewing the current locations of SQL Server databases

The Where to Move Databases screen contains the following information.

Databases on local disks: The list in the upper left portion of the screen displays the names of the SQL Server databases that reside on local disks. A database would appear in this list for one of the following reasons:

- The database does not reside on LUNs.
- The database resides on LUNs, but it does not meet the requirement for operating with SnapManager.
- Some of the files of the database reside on LUNs, and other files of the database reside on local disks.

Note

To expand a database in order to expose the data files and transaction log file, click the "+" next to the database name.

LUNs on the storage system: The list in the upper right portion of the screen lists the LUNs on the storage system. For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN.

Databases on LUNs: The list in the lower left portion of the screen lists the SQL Server databases that have been migrated to LUNs. A database would appear in this list for one of the following reasons:

- The database has been migrated to the LUN using the SnapManager Configuration Wizard.
- The database has been created on the LUN or moved to the LUN using some other SQL Server database utility, such as SQL Server Enterprise Manager or SQL Server Management Studio, and the configuration meets the requirements for operating with SnapManager.

Note

To expand a database in order to expose the data files and transaction log files, click the "+" next to the database name.

Migrating SQL Server databases from local disk to LUNs

To move SQL Server databases from a local disk to LUNs, complete the following steps from within the Where to Move Databases screen of the Configuration Wizard.

Step	Action
1	If you plan to place the SQL Server transaction log files on the same LUNs as the SQL Server data files, go to Step 4.
Preparing to place data files and log files on different LUNs.	
2	Check that there are enough LUNs available for you to place the data files and transaction log files on different LUNs. If there are not enough LUNs available, then do the following: **a.** Click Cancel to exit the Configuration Wizard. **b.** Use SnapDrive to create new LUNs. **c.** Re-attempt the SQL Server database migration.
3	Go to the list in the upper left pane and expand the view for the database you want to migrate. This is done by clicking the "+" next to the database name.

Assign databases to a LUN.	
4	In the upper list, select the databases (or individual data files or transaction log files) that you want to migrate. **Note** If an SQL Server 2005 database has the full-text search option enabled, the *full-text search catalogs* are visible when you click the "+" next to the database name. The full text catalogs can be migrated, backed up, and restored along with the other files or filegroups of the database. **To select a single database:** Click to select the database name. **To select a range of databases:** Select the first database in the range and then press and hold the Shift key on your keyboard while you select the last database in the range. **To select multiple databases:** Select the first database and then press and hold the Ctrl key on your keyboard while you select additional databases. **To place data files and log files on different LUNs:** (for a configuration consisting of one or more databases sharing two LUNs, create two mappings. ◆ In one mapping, select only data files and map them to a LUN. ◆ In the other mapping, select only transaction log files and map them to a different LUN. **Note** You will need repeat this sequence (Step 4 through Step 6) in order to create the second mapping. To select multiple files, select the first file and then press and hold the Ctrl key on your keyboard while you select additional files.

5	In the Disk List, select the LUN to which you want to migrate the selected databases or files. **Note** For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN.
6	Click the move (<=>) button. **Result:** The lower list is updated to include the new mapping.
If you need to undo any database-LUN assignments:	
7	The Where to Move Databases screen supports two types of undo operations: ◆ To undo a particular mapping in the "Database valid/change location..." list, select the mapping in the lower list and then click Reconfigure. The selected database (or individual data file or transaction log file) is returned to the list of unconfigured databases in the upper list. ◆ To undo all changes made during this session in the Where to Move Databases screen, click Undo All.

Moving SQL Server databases to different LUNs

To change the location of a database already associated with a LUN, complete the following steps from within the Where to Move Databases screen of the Configuration Wizard.

Step	Action
1	If you plan to move entire databases (as opposed to moving the SQL Server transaction log files separately from the SQL Server data files), go to Step 4.

Preparing to move log files separately from the data files	
2	Check that there are enough LUNs available for you to move the transaction log files separately from the data files. If there are not enough LUNs available, then do the following: **a.** Click Cancel to exit the Configuration Wizard. **b.** Use SnapDrive to create new LUNs. **c.** Re-attempt the SQL Server database migration.
3	Expand both of the database lists so that the data files and transaction log files are visible. **a.** Go to the upper left list and expand the view for the database you want to migrate. This is done by clicking the "+" next to the database name. **b.** Go to the lower left list and expand the view in order to expose the data files and transaction log files in their present locations. **Result:** Now you can associate the data files and transaction log files individually with a LUN.
Move databases to different LUNs	
4	In the lower left list, highlight the mapping you want to change, then click Reconfigure. **Result:** The mapping is removed from the lower left list. The database (or individual data file or transaction log file) is returned to the upper left list, and you can reassociate it with any available LUN.

5	In the upper left list, select the database (or individual data file or transaction log file) that you just removed from the lower left list. **To select a single database:** Click to select the database name. **To select a range of databases:** Select the first database in the range and then press and hold the Shift key on your keyboard while you select the last database in the range. **To select multiple databases:** Select the first database and then press and hold the Ctrl key on your keyboard while you select additional databases. **To place data files and log files on different LUNs:** For a configuration consisting of one or more databases sharing two LUNs, create two mappings. ◆ In one mapping, select only data files and map them to a LUN. ◆ In the other mapping, select only transaction log files and map them to a different LUN. **Note** You will need repeat this sequence (Step 5 through Step 7) in order to create the second mapping. To select multiple files, select the first file and then press and hold the Ctrl key on your keyboard while you select additional files.
6	In the Disk List, select the LUN to move it to.
7	Click the move (<=>) button. **Result:** The lower left list is updated to include the newly specified mapping.

Viewing the full path for a database on a LUN

To view the full path for a database (or individual data file or transaction log file) that has been migrated to a LUN, complete the following steps from within the Where to Move Databases screen of the Configuration Wizard.

Step	Action
1	If you plan to view the path of an entire database (as opposed to an individual data file or transaction log files), go to Step 3.
2	Go to the lower left list and expand the view in order to expose the data files and transaction log files. **Result:** Now you can view the data files and transaction log files individually with a LUN.
3	In the lower left list, select the database component and then click Details. **Result:** The Database Properties dialog box opens and displays the path information

Using the Number of Databases that Can Be Backed Up Concurrently screen

Use the Number of Databases that Can be Backed Up Concurrently screen to view and change the maximum number of databases that can be backed up concurrently. For more information, go to any of the following topics:

- "Implications of the maximum number of databases that can be backed up concurrently" on page 417
- "Changing the maximum number of databases that can be backed up concurrently" on page 417

Related topics:

- "SQL Server configuration rules with SnapManager" on page 86
- "SQL Server configurations supported with SnapManager" on page 89
- "Migrating SQL Server databases to LUNs" on page 107
- "Using the Modify SQL Server Maximum Worker Threads Parameter screen" on page 419

Implications of the maximum number of databases that can be backed up concurrently

By default, SnapManager backs up all databases on the same storage system volume concurrently. To limit the system resource used by SQL Server during SnapManager Snapshot copy backup, you can limit the number of databases that can be backed up concurrently.

Note

Limiting the number of databases can be backed up concurrently may increase the number of Snapshot copies that are required to backup all of the databases. In the case of multiple databases sharing one or two LUNs, if the number of databases allowed to be backed up concurrently is less than the actual number of databases that share the LUNs, then the backup will be split. In this case, a copy-based restore method will be used to restore from this backup set.

Changing the maximum number of databases that can be backed up concurrently

To change the maximum number of databases that can be backed up concurrently, complete the following steps from within the Number of Databases that Can Be Backed Up Concurrently screen of the Configuration Wizard.

Note

The maximum number of databases allowed per backup can also be configured using the Backup Setting dialog box. See “Configuring the profile of a full database backup” on page 378.

Step	Action
1	To change the maximum number of databases that can be backed up concurrently, type a new value in the text box labeled Maximum Databases to be Snapshot Backed up in Parallel. The default value is 255.
2	If you want to modify the maximum number of SQL Server worker threads, select the Modify SQL Server “Maximum worker threads” parameter option. This option is deselected by default.

Using the Modify SQL Server Maximum Worker Threads Parameter screen

Use the Modify SQL Server Maximum Worker Threads Parameter screen to view and change the maximum number of worker threads allowed per SQL Server instance. For more information, go to any of the following topics:

- "Calculating the maximum worker threads for an SQL Server instance" on page 419
- "Viewing the SQL Server maximum worker thread information" on page 419
- "Changing the maximum worker threads for an SQL Server instance" on page 420

Related topics:

- "SQL Server configuration rules with SnapManager" on page 86
- "SQL Server configurations supported with SnapManager" on page 89
- "Migrating SQL Server databases to LUNs" on page 107
- "Using the Number of Databases that Can Be Backed Up Concurrently screen" on page 417

Calculating the maximum worker threads for an SQL Server instance

SQL Server uses extra worker threads to perform a Snapshot-based database backup. To back up multiple databases on a single SQL Server instance concurrently, you might need to increase the maximum number of worker threads allowed for that SQL Server instance. If an SQL Server instance has more than 255 databases, you should increase the default setting.

Viewing the SQL Server maximum worker thread information

The Modify SQL Server Max Thread Option screen displays the following information about each SQL Server instance.

SQL Server Instance: This column displays the name of the SQL Server instance.

Current Max Worker Threads: This column displays the *current* maximum number of worker threads configured for this SQL Server instance.

New Max Worker Threads: This column displays the *recommended* maximum number of SQL Server worker threads for this SQL Server instance. This value is calculated based in part on the maximum number of databases that can be backed up concurrently.

The maximum number of databases that can be backed up concurrently is configured in a previous screen of the Configuration Wizard. For more information, see "Using the Number of Databases that Can Be Backed Up Concurrently screen" on page 417.

Changing the maximum worker threads for an SQL Server instance

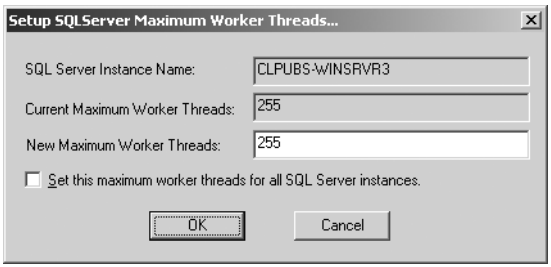
To change the maximum number of worker threads allowed by an SQL Server instance, complete the following steps from within the Modify SQL Server Maximum Worker Threads Parameter screen of the Configuration Wizard.

Note

If you choose to change the maximum number of worker threads for any SQL Server instances, those instances will be restarted when you click Finish in the Completing the Configuration Wizard screen.

Step	Action
1	Note the values displayed for each SQL Server instance in the list. ◆ The Current Max Worker Threads column of the this screen displays the *current* setting for each SQL Server instance. ◆ The New Max Worker Threads column of this screen displays the *recommended* setting for each SQL Server instance as determined by the maximum number of databases that can be backed up concurrently.
2	Before proceeding, note the "Apply new maximum worker thread value for each instance" option. This option is selected by default. If you leave this option selected when you click Next, the values in the New Max Worker Threads column will be applied to the SQL Server instances after you complete the Configuration Wizard sequence. Every affected SQL Server instance will be restarted in order for the new parameter value to take effect.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="4">3</td><td colspan="2">Choose whether you want to modify the maximum SQL Server worker threads as indicated by the values in the New Max Worker Threads column.</td></tr>
<tr><td>If...</td><td>Then...</td></tr>
<tr><td>You want to leave the current values unchanged for all of the SQL Server instances</td><td>1. Deselect the “Apply new maximum worker thread value for each instance” option.
2. Click Next.
Result: The SQL Server values will not be modified, and you have finished this procedure; the SnapInfo Directory Type screen appears.</td></tr>
<tr><td>You want to apply the recommended values for all of the SQL Server instances</td><td>1. Leave the “Apply new maximum worker thread value for each instance” option selected
2. Click Next.
Result: The values will be modified, and you have finished this procedure; the SnapInfo Directory Type screen appears.</td></tr>
<tr><td>You want to edit the recommended value for one or more of the SQL Server instances before applying the changes</td><td>1. Leave the “Apply new maximum worker thread value for each instance” option selected.
2. Proceed to Step 4.</td></tr>
</table>

Step	Action
4	Select the SQL Server instance whose recommended maximum worker thread count you want to modify, then click Edit New Value. **Result:** The Set Up SQL Server Maximum Worker Threads dialog box appears. 
5	In the New Maximum Worker Threads box, enter the new value.
6	You can apply this change to all SQL Server instances, or you can make this change individually for other SQL Server instances. ◆ If you want to apply this change to all SQL Server instances, select the option labeled "Set the maximum worker threads for all SQL Server instances." ◆ If you want to modify the maximum worker thread count for another individual SQL Server instance, go to Step 1.
7	Click OK to close the dialog box. **Result:** In the Modify SQL Server Maximum Worker Threads Parameter screen, the value New Max Worker Threads column is updated to reflect the new setting you specified. **Note** The instances whose values you modified will be restarted when you click Finish in the Completing the Configuration Wizard screen.
8	If you want to modify the maximum SQL Server worker threads for another SQL Server instance, repeat Step 4 through Step 7.

Using the Specify a Single SnapInfo Directory screen

Use the Specify a Single SnapInfo Directory screen to create one SnapInfo directory for all SQL Server instances and their associated databases.

Related topics:

- "SQL Server configuration rules with SnapManager" on page 86
- "How databases are stored on storage system volumes" on page 102
- "Migrating SQL Server databases to LUNs" on page 107
- "How SnapManager backup data is organized" on page 139
- "Using the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen" on page 425

To create a default SnapInfo directory for all the databases in an SQL Server instance, complete the following steps from within the Specify a Single SnapInfo Directory screen of the Configuration Wizard.

Step	Action
1	In the Disk List, locate the LUN on which you want the single SnapInfo directory to reside. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see "SQL Server database configuration restrictions enforced" on page 87.

Step	Action
2	Use any of the following methods to specify the location for the SnapInfo directory: ◆ Select the LUN in the Disk List, and then click the move (<=>) button. ◆ Double-click the LUN in the Disk List. ◆ Type the full path name of the SnapInfo directory in the Result SnapInfo Directory box. **Note** For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN. **Result:** The Result SnapInfo Directory box displays the path for the SnapInfo directory. Note that the default directory name is `SMSQL_SnapInfo`.
3	If you want to specify a different location or name, modify the information in the Result SnapInfo Directory box. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see "SQL Server database configuration restrictions enforced" on page 87.

Using the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen

Use the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen to create a default SnapInfo directory for all the databases in an SQL Server instance.

Related topics:

- "Using the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen" on page 428
- "Using the Where to Put the SnapInfo Directory for Each Database screen" on page 431
- "SQL Server configuration rules with SnapManager" on page 86
- "How databases are stored on storage system volumes" on page 102
- "Migrating SQL Server databases to LUNs" on page 107
- "How SnapManager backup data is organized" on page 139

To create a default SnapInfo directory for all the databases in an SQL Server instance, complete the following steps from the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen of the Configuration Wizard.

Note

Later, in the Where to Put the SnapInfo Directory for Each Database screen, you will have the opportunity to override this setting for any individual database.

Step	Action
1	In the unlabeled list, expand an SQL Server instance.
2	In the Disk List, select the LUN on which you want to create the default SnapInfo directory for this instance. **Note** For more information about any LUN in the Disk List, move the mouse over the LUN name. This causes a tooltip to appear and display the IP address or host name of the storage system and the name of the volume that houses the LUN.

Step	Action
3	Click the move (<=>) button. **Result:** The Default SQL Server SnapInfo Directory Path List displays the path for the SnapInfo directory. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see "SQL Server database configuration restrictions enforced" on page 87.
4	If you want to unmap the SnapInfo directory for an SQL Server instance, do the following: **a.** In the Default SQL Server SnapInfo Directory Path List, select the SQL Server instance. **b.** Click Reconfigure. **Result:** The LUN is removed from the Default SQL Server SnapInfo Directory Path List. The LUN is returned to the list of unconfigured database LUNs, and you can reassociate its SnapInfo directory with any available LUN.

Step	Action
5	By default, the SnapInfo directory name is `SMSQL_SnapInfo`. If you want to specify a different location or name, complete the following steps: **a.** In the Default SQL Server SnapInfo Directory Path List, select the SQL Server instance for which you want to change the location or name of the SnapInfo directory. **b.** Click Details. The Default SnapInfo Property dialog box appears and displays the current location specified for the SnapInfo directory for the selected SQL Server instance. **c.** In the New SnapInfo Directory Path box, enter the new path. **d.** Click OK. The dialog box closes, and the To column of the Default SQL Server SnapInfo Directory Path List displays the updated path information. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see "SQL Server database configuration restrictions enforced" on page 87.

Using the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen

If you have multiple databases that reside on one or two LUNs, use the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen to specify a common SnapInfo directory for those databases.

Related topics:

- "Using the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen" on page 425
- "Using the Where to Put the SnapInfo Directory for Each Database screen" on page 431
- "SQL Server configuration rules with SnapManager" on page 86
- "How databases are stored on storage system volumes" on page 102
- "Migrating SQL Server databases to LUNs" on page 107
- "How SnapManager backup data is organized" on page 139

To create a common SnapInfo directory for multiple databases that share one or two LUNs, complete the following steps from the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen of the Configuration Wizard.

Step	Action
1	In the unlabeled list, select the LUN or LUNs that you want to map to a shared SnapInfo directory
2	In the Disk List, select the LUN to move it to.
3	Click the move (<=>) button. **Result:** The Virtual Disk SnapInfo Directory Path List displays the updated path information. The SnapInfo directory for the databases sharing the selected LUN or LUNs will be stored at this location rather than at the default location for the SQL Server instance. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see "SQL Server database configuration restrictions enforced" on page 87.

Step	Action
4	If you want to unmap the SnapInfo directory for multiple databases that share one or two LUNs, do the following: **a.** In the Virtual Disk SnapInfo Directory Path List, select the shared LUN. **b.** Click Reconfigure. **c.** If the databases share a second LUN, repeat the preceding steps. **Result:** The LUN is removed from the Virtual Disk SnapInfo Directory Path List. The LUN is returned to the list of unconfigured database LUNs, and you can reassociate its SnapInfo directory with any available LUN.

Step	Action
5	By default, the SnapInfo directory name is `SMSQL_SnapInfo`. If you want to specify a different location or name, complete the following steps: **a.** In the Virtual Disk SnapInfo Directory Path List, select the LUN or LUNs for which you want to change the location or name of the SnapInfo directory. **b.** Click Details. The Default SnapInfo Property dialog box appears and displays the current location specified for the SnapInfo directory for the databases that share the selected LUN or LUNs. **c.** In the New SnapInfo Directory Path box, enter the new path. **d.** Click OK. The dialog box closes, and the To column of the Virtual Disk SnapInfo Directory Path List displays the updated path information. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see "SQL Server database configuration restrictions enforced" on page 87.

Using the Where to Put the SnapInfo Directory for Each Database screen

In the Where to Put the SnapInfo Directory for Each Database screen, you specify a default SnapInfo directory location to be used by *all* the databases in a selected SQL Server instance. If you want to specify a different SnapInfo directory location for an *individual* database, use the Where to Put the SnapInfo Directory for Each Database screen to specify the location for that SnapInfo directory.

Related topics:

- "Using the Where to Put the Default SnapInfo Directory for Each SQL Server Instance screen" on page 425
- "Using the Where to Put the Default SnapInfo Directory for LUNs Shared by Multiple Databases screen" on page 428
- "SQL Server configuration rules with SnapManager" on page 86
- "How databases are stored on storage system volumes" on page 102
- "Migrating SQL Server databases to LUNs" on page 107
- "How SnapManager backup data is organized" on page 139

To create a SnapInfo directory for a specific database, complete the following steps from the Customize SnapInfo Directory (Per Database) screen of the Configuration Wizard.

Step	Action
1	In the unlabeled list, select the individual database that you want to map to a dedicated SnapInfo directory.
2	In the Disk List, select the LUN on which you want to create the SnapInfo directory for the database.

Step	Action
3	Click the move (<=>) button. **Result:** The Database SnapInfo Directory Path List displays the location at which the SnapInfo directory will be created. The SnapInfo directory for the selected database will be stored at this location rather than at the default location for the SQL Server instance. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see "SQL Server database configuration restrictions enforced" on page 87.
4	If you want to unmap the SnapInfo directory for an SQL Server database, do the following. **a.** In the Database SnapInfo Directory Path List, select the SQL Server database. **b.** Click Reconfigure. **Result:** The mapping is removed from the Database SnapInfo Directory Path List. The SQL Server instance is returned to the list of unconfigured SQL Server databases, and you can reassociate its SnapInfo directory with any available LUN.

Step	Action
5	By default, the SnapInfo directory name is `SMSQL_SnapInfo`. If you want to specify a different location or name, complete the following steps. **a.** In the Database SnapInfo Directory Path List, select the database for which you want to change the location or name of the SnapInfo directory. **b.** Click Details. The Default SnapInfo Property dialog box appears and displays the current location specified for the SnapInfo directory for the selected database. **c.** In the New SnapInfo Directory Path box, enter the new path. **d.** Click OK. The dialog box closes, and the To column of the Database SnapInfo Directory Path List displays the new location for the SnapInfo directory for the selected database. **Note** The Configuration Wizard will allow you to create the SnapInfo directory only in valid locations. For more information, see “SQL Server database configuration restrictions enforced” on page 87.

Using the Database Migration Options screen

Use the Database Migration Options screen to specify when you want to run the DBCC command to verify your migrated databases. Additionally, you can choose to delete the original database files after the migration operation finishes successfully.

Related topics:

- "How SnapManager checks database integrity in backup sets" on page 135
- "Migrating SQL Server databases to LUNs" on page 107
- "Using the Database Verification Server and Account Authentication screen" on page 408

To specify database migration options, complete the following steps from the Database Migration Options screen of the Configuration Wizard.

Step	Action
1	Select the options that specify when you want to run the DBCC command to verify your migrated databases: **a.** If you want to verify the databases before the migration begins, select the Run DBCC Physical Integrity Verification Before Migration option. This option is selected by default. **b.** If you want to verify the databases after the migration finishes, select the Run DBCC Physical Integrity Verification After Migration option. This option is deselected by default.
2	If you want to delete the original database files after the migration operation finishes successfully, select the Delete Copy of Migrated Database at Original Location option. This option is selected by default.
3	If you want to omit updating statistics on all tables before detaching the databases, deselect the "Run UPDATE STATISTICS on tables before detaching the databases" option. This option is selected by default.

Configuring Post-Restore Database Recovery

E

When you specify a database restore operation, you have several choices for the states in which you want the restored databases to be left after the operation finishes. This section describes these choices. The following topics are covered:

- "Understanding post-restore database recovery states" on page 436
- "Specifying the post-restore state of a single database" on page 437
- "Specifying the post-restore state of multiple databases" on page 441

Related topics:

- "Restoring using the SnapManager Restore Wizard" on page 236
- "Restoring using the SnapManager Restore tab" on page 229

Understanding post-restore database recovery states

The following table describes the post-restore database states from which you can choose.

Database state	Description
Operational	All of the following apply: ◆ No additional transaction logs can be restored. ◆ The database is ready to use. This database state is selected by default.
Non-Operational	Additional transaction logs can be restored.
Read-Only	All of the following apply: ◆ Additional transaction logs can be restored. ◆ The undo file is enabled. If additional transaction logs are restored, any changes can be rolled back in case the restoration of the transaction log is unsuccessful. **Note** If you restore a database to a temporary, alternate location using a writable Snapshot copy with this option enabled, the Detach Database and Dismount Snapshot LUN(s) function is unavailable for this database.

Specifying the post-restore state of a single database

When specifying a single-database restore operation, you can choose the state that you want the database to be left in after the restore operation finishes. See the following topics for more information:

- "Specifying single-database recovery state from the SnapManager Restore tab" on page 437
- "Specifying single-database recovery state from within the Restore Wizard" on page 438

If you are performing a multiple-database restore operation, see "Specifying the post-restore state of multiple databases" on page 441.

Related topics:

- "Understanding SnapManager Restore" on page 216
- "Performing a restore operation" on page 225

Specifying single-database recovery state from the SnapManager Restore tab

When using the SnapManager Restore tab to restore a single database, you use the Restore Option dialog box to specify the state in which the database is to be left after the restore operation finishes.

To specify the database post-restore state for a single-database restore operation started using the SnapManager Restore tab, complete the following steps.

Step	Action
Open the Restore Option dialog box.	
1	From the SnapManager Restore tab, open the Restore Option dialog box. **a.** Specify a single-database restore operation. This is described in "Restoring using the SnapManager Restore tab" on page 229. **b.** As described in Step 14 of that procedure, click Restore when you are ready to start the restore operation. **Result:** The Restore Option dialog box appears and prompts you to select the post-backup state for the database.

Step	Action
Specify the post-restore database state.	
2	Select the state that you want the database to be left in after the restore operation finishes: ◆ Leave database operational, but unavailable for restoring additional transaction logs. ◆ Leave database nonoperational, but available for restoring additional transaction logs. ◆ Leave database in read-only mode and available for restoring additional transaction logs. For descriptions of these options, see "Understanding post-restore database recovery states" on page 436.
3	If you selected "Leave database read-only and able to restore additional transaction logs," you must also specify the undo file. You can either type the file name in the Undo File box or click "..." to browse to the file.
4	Click OK to close the dialog box.
Start the single-database restore operation.	
5	Continue with Step 15 in the procedure described in "Restoring using the SnapManager Restore tab" on page 229.

Specifying single-database recovery state from within the Restore Wizard

When using the Restore Wizard to restore a single database, you use the Database State After the Restore screen to specify the state that you want the database to be left in after the restore operation finishes.

To specify the database post-restore state for a single-database restore operation started using the Restore Wizard, complete the following steps.

Step	Action
Open the Database State After the Restore screen.	
1	Step through the Restore Wizard screens, specifying a single-database restore operation, until you reach the Database State After the Restore screen. This is described in "Restoring using the SnapManager Restore Wizard" on page 236. Step 6 of that procedure lists the names of the Restore Wizard screens. **Result:** The Database State After the Restore screen prompts you to choose the post-backup state for the databases.
Specify the post-restore database state.	
2	In the Database State After the Restore screen, select the state that you want the database to be left in after the restore operation is finished: ◆ Leave database operational, but unavailable for restoring additional transaction logs. ◆ Leave database nonoperational, but available for restoring additional transaction logs. ◆ Leave database in read-only mode and available for restoring additional transaction logs. For descriptions of these options, see "Understanding post-restore database recovery states" on page 436.
3	If you selected the "Leave database nonoperational..." option or the "Leave database read-only..." option, you must also specify the undo file. You can either type the file name in the Undo File box or click "..." to browse to the file.
4	To apply your settings and proceed to the next screen of the Restore Wizard, click Next. **Result:** The Restore wizard displays the Restored Database Name screen.

Step	Action
Finish specifying the single-database restore operation.	
5	Continue with Step 6 in the procedure described in "Restoring using the SnapManager Restore Wizard" on page 236.

Specifying the post-restore state of multiple databases

When specifying a multiple-database restore operation, you can choose the states that you want each of the databases to be left in after the restore operation finishes. See the following topics for more information:

- "Specifying multiple-database recovery state from the SnapManager Restore tab" on page 441
- "Specifying multiple-database recovery state from within the Restore Wizard" on page 443

If you are performing a single-database restore operation, see "Specifying the post-restore state of a single database" on page 437.

Related topics:

- "Understanding SnapManager Restore" on page 216
- "Performing a restore operation" on page 225

Specifying multiple-database recovery state from the SnapManager Restore tab

When using the SnapManager Restore tab to restore multiple databases, you use the Multiple Databases Restore Option dialog box to specify the states in which the databases are to be left after the restore operation finishes.

To specify database recovery states for a multiple-database restore operation started using the SnapManager Restore tab, complete the following steps.

Step	Action
Open the Multiple Databases Restore Options dialog box.	
1	From the SnapManager Restore tab, open the Multiple Databases Restore Options dialog box. **a.** Specify a multiple-database restore operation. This is described in "Restoring using the SnapManager Restore tab" on page 229. **b.** As described in Step 14 of that procedure, click Restore when you are ready to start the restore operation. **Result:** The Multiple Databases Restore Options dialog box appears and prompts you to choose the post-backup state for the databases.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Specify the post-restore database state.</td></tr>
<tr><td>2</td><td>Select the state that you want the database to be left in after the restore operation finishes.
Leave databases operational. No additional transaction logs can be restored.
Leave databases nonoperational but able to restore additional transaction logs.
Leave databases read-only and able to restore additional transaction logs.
For descriptions of the database recovery states, see “Understanding post-restore database recovery states” on page 436.<table><tr><th>If...</th><th>Then...</th></tr><tr><td>All the databases are to be operational</td><td>Leave the “Leave databases operational...” option selected.</td></tr><tr><td>All of the databases are to be nonoperational</td><td>Select the “Leave databases nonoperational...” option.</td></tr><tr><td>Some of the databases are to be operational, and other databases are to be nonoperational</td><td>1. Select the “Leave databases operational...” option.
2. In the database list in the middle of the dialog box, deselect any databases that are to be operational.</td></tr><tr><td>All the databases are to be read-only</td><td>Select the “Leave databases read-only...” option.</td></tr></table></td></tr>
<tr><td>3</td><td>If you selected the “Leave databases nonoperational...” option or the “Leave databases read-only...” option, you must also specify the directory that contains the undo file.
You can either type the directory name in the Undo File box or click “...” to browse to the directory.</td></tr>
</table>

Step	Action	
Confirm restore options.		
4	Click to activate the Confirm Restore Options Selected tab. **Result:** The Confirm Restore Options Selected tab displays the restore options currently in effect.	
5	Review all of the restore options shown in the tab.	
	If you want to...	**Then...**
	Modify any of the currently specified restore options	Click Cancel to close the dialog box. Modify any restore options (as described in "Restoring using the SnapManager Restore tab" on page 229) before proceeding with the restore operation.
	Proceed with the restore operation with the currently specified restore options	Click OK to close the dialog box.
Start the multiple-database restore operation.		
6	Continue with Step 15 in the procedure described in "Restoring using the SnapManager Restore tab" on page 229.	

Specifying multiple-database recovery state from within the Restore Wizard

When using the Restore Wizard to restore multiple databases, you use the Database State After the Restore screen to specify the state that you want the database to be left in after the restore operation finishes.

To specify the database post-restore state for a multiple-database restore operation started using the Restore Wizard, complete the following steps.

Step	Action
Open the Database State After the Restore screen.	
1	Step through the Restore Wizard screens, specifying a multiple-database restore operation, until you reach the Database State After the Restore screen. This is described in "Restoring using the SnapManager Restore Wizard" on page 236. Step 6 of that procedure lists the names of the Restore Wizard screens. **Result:** The Database State After the Restore screen prompts you to choose the post-backup state for the databases.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Specify the post-restore database state.</td></tr>
<tr><td>2</td><td>Select the state that you want the database to be left in after the restore operation finishes.
Leave databases operational. No additional transaction logs can be restored.
Leave databases nonoperational but able to restore additional transaction logs.
Leave databases read-only and able to restore additional transaction logs.
For descriptions of the database recovery states, see “Understanding post-restore database recovery states” on page 436.<table><tr><th>If...</th><th>Then...</th></tr><tr><td>All the databases are to be operational</td><td>Leave the “Leave databases operational...” option selected.</td></tr><tr><td>All of the databases are to be nonoperational</td><td>Select the “Leave databases nonoperational...” option.</td></tr><tr><td>Some of the databases are to be operational, and other databases are to be nonoperational</td><td>1. Select the “Leave databases operational...” option.
2. In the database list in the middle of the dialog box, deselect any databases that are to be operational.</td></tr><tr><td>All the databases are to be read-only</td><td>Select the “Leave databases read-only...” option.</td></tr></table></td></tr>
<tr><td>3</td><td>If you selected the “Leave databases nonoperational...” option or the “Leave databases read-only...” option, you must also specify the directory that contains the undo file.
You can either type the directory name in the Undo File box or click “...” to browse to the directory.</td></tr>
</table>

Step	Action
4	To apply your settings and proceed to the next wizard screen, click Next. **Result:** The Restore Wizard displays the Restored Database Name screen.
Finish specifying the single-database restore operation.	
5	Continue with Step 6 in the procedure described in "Restoring using the SnapManager Restore Wizard" on page 236.

Index

Symbols

*.bak files 132
*.fbk files 132
*.trb files 134
*.trn files 134
*__Mount backup set name 217
*__recent Snapshot names 143

A

Access LUN in Snapshot 376
administration, Microsoft SQL Server xi
archives
    disaster recovery guidelines 299
    recovering SQL Server databases 309
    restoring databases from 309
    restoring volumes from SnapManager backups archived with SnapVault 314
archiving SnapManager backup sets
    about
        choosing the best way to archive 270
        guidelines for 269
        importance of complete backup sets 268
        scheduling considerations for 269
    initiated automatically
        after a backup or verification 340
        using SnapVault 278
    initiated manually
        using a Windows backup utility 274
        using NDMP or dump 271
    unsupported methods
        using CIFS 269
        using NFS 269
authentication method
    SQL Server 26, 28, 349
    Windows 26
Auto Shrink option, SQL Server database 186

B

Backup and Verification tab
    Invalid database label 130, 132
    performing a database verification 189
    performing a full database backup 157
    performing a transaction log backup 175
    scheduling a job to run later 349
    using the Run Command After Operation feature 340
backup management groups
    about 198
    assigning a new full database backup to a group 198
    changing the group assignment for an existing full database backup 200
    using with SnapManager operations
        database verification 187, 199
        explicit deletion of multiple backup Snapshot copies 199, 203
        full database backup 144, 152, 199
backup method
    Snapshot-based 133
    stream-based 132
backup retention. *See* Snapshot copies
backup sets, SnapManager
    archiving a complete backup set 268
    data organization within 139
    guidelines for restoring 225
    how Snapshot copies are used 17
    labeled SnapLUN 217, 244, 245
    naming convention for 141
Backup Wizard
    Invalid database label 130, 132
    performing a database verification 193
    performing a full database backup 162
    performing a transaction log backup 179
    scheduling a job to run later 349
    using the Run Command After Operation feature 340
bulk-logged recovery model, SQL Server
    as supported by SnapManager 214
    definition of 10
busy Snapshot
    avoiding after a SnapManager operation
        mounting and attaching a database

without restoring it 217, 243, 245
restoring to an alternate location using writable Snapshot 217, 245
avoiding during a SnapManager operation
database verification only 188, 189
full database backup with verification 155
when leaving database attached after verification 375
avoiding while archiving SnapManager backups
using a Windows backup utility 275
using NDMP or `dump` 272
using SnapVault 286
avoiding while restoring a volume from a SnapVault archive 318
deleting
unable to delete using SnapManager 210
using Data ONTAP 211

C

CIFS protocol, as supported by SnapManager
access to LUN objects 322
cannot be used to archive LUNs 269
cannot be used to back up or restore databases 5
cluster. *See* Windows cluster
color, database icon 157, 162
Configuration Wizard
about 103
how it stores databases on volumes 102
migrating databases back to local disks 126
migrating databases from local disks to LUNs 107
moving multiple SnapInfo directories to a single SnapInfo directory 123
`Reconfig` database label 126
when to run the Configuration Wizard 104
copy-based restore method 219

D

Data ONTAP
compatibility with SnapDrive 21
product documentation xi, xii
storage system requirement 31
database consistency checker. *See* DBCC
database icon color 157, 162
database label
`Invalid` 130, 132
`Reconfig` 126
database verification
avoiding busy Snapshot 375
information you need to specify 187
running a command after the operation finishes 340
scheduling the job to run later 349
using the Backup and Verification tab 189
using the Backup Wizard 193
databases, SQL Server
backing up after renaming 225
backing up before installing SnapManager 33
maximum per LUN 86
maximum per SQL Server computer 86
maximum per SQL Server instance 86
restoring from archive 309
DBCC
as used by SnapManager Backup 135, 188, 211
as used by SnapManager Restore 221, 228, 242
as used by the Configuration Wizard 105, 106, 119, 434
as used by the `SmSqlBi` command (the CLI) 360, 361
definition of 9
drive letters required for verifying a backup 136
specifying settings 155, 374
disaster recovery
general guidelines 299
restoring system databases 337
using NDMP or `dump` archives
general procedure 309
guidelines 299
using other SQL Server backup sets 326
using SnapMirror replication
general procedure 301
guidelines 298
using SnapVault archives
general procedure 314
guidelines 299
distribution database, definition of 12

documentation
NetApp on the Web (NOW) 21
reference documents 20
drive letters required for DBCC 136
dump command, storage system
archiving LUNs that contain SnapManager backup sets 3, 16, 269, 271
compared with other archive methods 270

E

Enterprise Manager.
See SQL Server Enterprise Manager

F

FCP
LUN access protocol 25, 32
LUN driver 29
filer. *See* storage system
filer-side license.
See per-storage system license
Fractional space reservation policy settings in SMSQL 388
full database backup
failure of concurrent backup of multiple databases 168
information you need to specify 152
running a command after the operation finishes 340
scheduling the job to run later 349
selecting databases at the instance level
about 152
using the Backup and Verification tab 157
using the Backup Wizard 162
Snapshot copy-based backup method 133
stream-based backup files 132
stream-based backup method 132
using the Backup and Verification tab 157
using the Backup Wizard 162
volume-wide backups 133
what to do if the backup fails 184
full recovery model, SQL Server
as supported by SnapManager 214
definition of 10
full-text search catalogs, SQL Server
database 88, 412

G

guidelines
for archiving SnapManager backup sets 269
for disaster recovery 299
for disaster recovery using archives 299
for performing a SnapManager Restore operation
choosing the type of restore to perform 224
preparing to perform a restore operation 225
for restoring from a SnapManager backup set 225
for volume sizing 99

I

installation directory, SnapManager
specifying during interactive installation 40
specifying during unattended installation 44
installing SnapManager on a standalone system
in interactive mode 37
in unattended mode 42
storage system requirements for 31
Windows host system requirements for 22
installing SnapManager on a Windows cluster
and creating a new Windows cluster 49
disk requirements for 48
storage system requirements for 31
system configuration requirements for 47
Windows host system requirements for 22
interactive mode
installing SnapManager 37
uninstalling SnapManager 60
upgrading SnapManager 52
Invalid database label 130, 132
IP address, setting preferred 31
iSCSI
LUN access protocol 25, 32
LUN access protocol documentation xii
LUN driver 29

L

label, database
    Invalid 130, 132
    Reconfig 126
license requirements
    FCP protocol 31
    iSCSI protocol 31
    SnapMirror 31
    SnapRestore 31
licenses
    SnapManager 26
    storage system requirements 31
    Windows host system requirements 25
log shipped databases
    database restore of 134, 228, 229, 382
    transaction log backup of 134
    up-to-the-minute restore of 222
LUN access protocol 25, 32
    iSCSI documentation xii
LUN driver 29
LUN size calculations 97
LUNs not visible in Configuration Wizard 112

M

management groups.
    See backup management groups
Management Studio.
    See SQL Server Management Studio
master database, definition of 12
MDAC (Microsoft Data Access Components) version 23, 69, 186, 369
Microsoft SQL Server. *See* SQL Server
mirrored volumes
    recovering SQL Server databases from. *See* SnapMirror replication, restoring from
    replicating SnapManager backups to. *See* SnapMirror replication of SnapManager backups
model database, definition of 12
mount point
    limitations 74
    limitations in a clustered environment 74
Mounted volume naming conventions 79
mounted volumes in SnapManager 79
MSCS environment. *See* Windows cluster
msdb database, definition of 12
multiple-instance cluster 47

N

NDMP-based backup utility
    archiving LUNs that contain SnapManager backup sets 3, 16, 269, 271
    compared with other archive methods 270
NearStore, archiving LUNs that contain SnapManager backup sets 16
NFS protocol
    cannot be used to archive LUNs 269
NOW (NetApp on the Web) 21
NTBackup 16
NTFS 18
NVRAM 18

O

online Snapshot. *See* snapshot-based
operating system
    storage system requirements 31
    Windows host system requirements 23

P

per-SQL Server license 25, 26
per-storage system license 25, 26
point-in-time restore 223, 224
preferred IP address, setting 31
preinstallation
    preparing a Windows host system 33
    verifying storage system requirements 31
    verifying Windows host system requirements 22
    what to read 20
protocol
    CIFS 5, 269, 322
    FCP 25, 32
    iSCSI 25, 32
    NDMP 3, 16, 269, 271

R

rebuildm.exe (rebuild master) 337
`Rebuildm.exe` utility 337
`Reconfig` database label 126
recovery models, Microsoft SQL Server
    as supported by SnapManager 214
    definition of 10
remote administration server 22
    Microsoft SQL Server requirement for 24
    other requirements for 28
    other requirments for 25
    SnapDrive requirement for 24, 27
remote verification server
    connectivity requirements for 30
    drive letters required for DBCC 136
    other requirements for 29
    SnapDrive requirement for 27, 28
    system requirements 22
replicating SnapManager backups.
    See SnapMirror replication 255
Report directory, SnapManager
    changing the location 400
    default location 399
    option to remove during uninstallation 60, 62
    remote access to 68, 366, 368
reports, SnapManager operational
    deleting a report 253
    printing a report 252
    types of 250
    viewing a report 251
requirements
    storage system 31
    transaction log space 98
    Windows host system 22
restore method
    copy-based 219
    snapshot-based 219
    stream-based 219
restore Snapshot copies
    about 220
    deleting 247
restoring databases
    after renaming a database 225
    using a point-in-time restore 223
    using an up-to-the-minute restore 222
    with log-shipping implemented 134, 228, 229, 382
retention of backup Snapshot copies. *See* Snapshot copies
rolling snapshots, SnapDrive
    advantages over increased SnapManager backups 260
    to supplement automatic replication 259
Run Command After Operation feature
    and generic backup naming (`__recent` suffix) 142
    configuring default values 385
    running a script from a UNC path 347
    using 340

S

SAN boot LUN 88
scheduling
    archival of SnapManager backups 269
    running a backup or verification job for later 349
security authentication. *See* authentication method
server-side license. *See* per-SQL Server license
service account, SnapManager
    in workgroup mode 27, 41, 44
    requirements 26
setting a preferred IP address 31
`setup.exe` utility 337
simple recovery model, SQL Server
    as supported by SnapManager 214
    definition of 10
`SMSQLReportFolder` share 68, 366, 368
SnapDrive
    compatibility with SnapManager and Data ONTAP 21
    overview
        when to use as opposed to SnapManager 15
    product documentation xi, xii, 20, 21
    rolling Snapshots 259
    storage requirements 96
    Windows host system requirements 24
SnapInfo directory, SnapManager
    moving multiple SnapInfo directories to a

single SnapInfo directory 123
naming conventions for 140
rules for storing 102
SnapManager Backup 139
specifying the name and location
Advanced SnapInfo Directories 117
Single SnapInfo Directory 116
system databases stopped by SnapManager 102
transaction log backup file names 134
user databases detached by SnapManager 102
SnapLUN backup set label 217, 244, 245
SnapManager
application overview
command- line interface 8
graphical user interface 7
how it stores databases on volumes 102
how it uses Snapshot copies as backups 17
how it uses Snapshots as a restore precaution 220
how it works with other backup methods 16
how it works with SnapDrive 15
how you use it 6
maximum configurations supported by 86
new functionality with this version 2
relationship with other Data ONTAP-based components 13
relationship with SQL Server components 14
what it does 3
what it does not do 5
when to use it 15
when to use SnapDrive instead 15
where you install and run it 6
application settings
configurable from the Configuration Wizard 106
configurable outside of the Configuration Wizard 363
backup sets
archiving a complete backup set 268
data organization within 139
guidelines for restoring from 225
naming convention for 141
installing or upgrading
compatibility with SnapDrive 21
installing on a standalone system 36
installing on a Windows cluster 47
option to remove Report directory during uninstall 60, 62
preparing a Windows host system for 33
preparing to install 19
reinstalling 65
storage system requirements for 31
uninstalling 58
upgrading 51
what to read 20
Windows host system requirements for 22
terminology 9
user interface
command-line interface (CLI) 353
Configuration Wizard 101
SnapManager Backup 149
SnapManager Reports 249
SnapManager Restore 213
SnapManager Backup
deleting oldest Snapshot copies with 144
limitations 131
managing the number of Snapshot copies and backup sets 144
performing a database verification
about 135
default verification settings 372
information you need to specify 187
using the Backup and Verification tab 189
using the Backup Wizard 193
performing a full database backup
about 132
default backup settings 378
default verification settings 372
information you need to specify 152
using the Backup and Verification tab 157
using the Backup Wizard 162
performing a transaction log backup
about 134
default backup settings 378
information you need to specify 172
using the Backup and Verification tab 175
using the Backup Wizard 179

remote verification with 29
requirements 131
starting or scheduling jobs
database verification 187
full database backup 152
transaction log backup 172
using backup management groups 198
what it does 130
when to back up your databases 147
See also backup sets
SnapManager Reports
about 250
deleting a report 253
printing a report 252
Report directory
changing the location 400
default location 399
option to remove during uninstall 60, 62
viewing a report 251
SnapManager Restore
about 216
cluster failure during 220
cluster group state during 24, 220
preparing to restore a backup set 225
restoring from archive with 310
Snapshot copies created as a precaution 220
SnapManager support 76
SnapManager system requirements
storage system 31
Windows host system 22
SnapMirror replication of SnapManager backups
how it works 256
overview 255
scheduling considerations 263
storage system license requrements 31
supplementing with rolling Snapshots 259
SnapMirror replication, restoring from
disaster recovery guidelines 298
recovering SQL Server databases 301
SnapRestore 32
Snapshot copies
backup Snapshot copies
about 17
automatically deleting the oldest 144
explicitly deleting multiple 199
naming conventions 142
maximum allowed per volume 144
restore Snapshot copies
about 220
explicitly deleting 247
naming convention 220
Snapshot copy-based
backup method 133
restore method 91, 219
technology provided by Data ONTAP 3
SnapVault archives
archiving SnapManager backup sets 278
restoring a storage system volume from 314
space requirements for transaction logs 98
SQL Browser service
no LUNs in Configuration Wizard 112
Windows host system requirements 24
SQL Server
`*.bak` files 132
`*.trn` files 134
administration xi
calculating database size 97
full-text search catalogs 88, 412
recovery models
as supported by SnapManager 214
definition of 10
SnapManager rules for storing databases 102
system databases
definition of 11
stopped by SnapManager 102, 108, 115
user databases
definition of 12
detached by SnapManager 102, 108, 115
Windows host system requirements 24
See also SQL Browser service
See also SQL Server 2000
See also SQL Server 2005
See also SQL Server authentication method
See also SQL Server database
See also SQL Server Enterprise Manager
See also SQL Server instance
See also SQL Server Management Studio
SQL Server 2000
`Rebuildm.exe` utility 337
verifying an SQL Server 2005 database 138

SQL Server 2005
MDAC version 23, 69
no user databases on root LUN 88, 132
`setup.exe` utility 337
SQL Browser service 24, 112
verifying an SQL Server 2000 database 138
SQL Server authentication method
described 26
for accessing SQL Server 26
limitations when connecting to a remote administration server 28
limitations when scheduling a remote verification server 349
SQL Server database
Auto Shrink option 186
SQL Server Enterprise Manager
backing up transaction logs in a SnapManager environment 16
detecting SnapManager transaction log backups 134
viewing SnapManager full database backup files 133
SQL Server instance
as a remote administration server
other requirements for 25, 28
SnapDrive requirement for 24, 27
verifying host system requirements 22
as a remote verification server
connectivity requirements for 30
drive letters required for DBCC 136
other requirements for 29
SnapDrive requirement for 28
verifying host system requirements 22
authentication method used to connect to 26
maximum databases per 86
maximum per SQL Server computer 86
selecting databases at the instance level
full database backup 152, 157, 162
transaction log backup 172, 175, 179
verifying host system requirements for 22
See also SQL Server
SQL Server Management Studio
backing up transaction logs in a SnapManager environment 16
detecting SnapManager transaction log backups 134
viewing SnapManager full database backup files 133
`SQL__*` SnapInfo subdirectory names 140
`sqlsnap__*` snapshot names 143
storage system
licenses 31
requirements for SnapManager 31
stream-based operations
backup file names 132
backup method 132
restore method 91, 219
system databases, Microsoft SQL Server
backing up 133
definition of 11
distribution database, definition of 12
master database, definition of 12
migrating to LUNs 102
model database, definition of 12
msdb database, definition of 12
restoring 337
stopped by SnapManager 102
tempdb database, definition of 12
system requirements for SnapManager
storage system 31
Windows host system 22

T

tempdb database, definition of 12
transaction log backup
information you need to specify 172
of a log-shipped database 134
running a command after the operation finishes 340
scheduling the job to run later 349
selecting databases at the instance level
about 172
using the Backup and Verification tab 175
using the Backup Wizard 179
SnapManager backup data 134
using the Backup and Verification tab 175
using the Backup Wizard 179
what to do if the backup fails 184
transaction logs

rules for storing 102
space estimation 98
volume requirements for 98

U

unattended mode
installing SnapManager 42
uninstalling SnapManager 61
upgrading SnapManager 55
uninstalling SnapManager
before you uninstall 58
in interactive mode 60
in unattended mode 61
option to remove Report directory 60, 62
upgrading SnapManager
converting VLD-type virtual disks to LUNs 51
Data ONTAP requirement 51
in interactive mode 52
in unattended mode 55
Microsoft SQL Server requirement 51
storage system requirememts for 31
Windows host system requirements for 22
up-to-the-minute restore 222, 224
user databases, Microsoft SQL Server
backing up 133
definition of 12
detached by SnapManager 102
migrating to LUNs 102

V

`VDisk__*` SnapInfo subdirectory names 140
Virtual Device Interface (VDI) 3
VLD-type virtual disks 51
volume size
assessing 96
guidelines 99
requirements for database files 96
requirements for transaction logs 98
transaction log sizing 98
volumes, storage system
maximum per single database 86
maximum per SQL Server computer 86
restoring from archive 314
volume-wide backups 133

W

Windows authentication method
described 26
for accessing SQL Server 26
Windows backup utility
archiving SnapManager backup sets 274
compared with NDMP or `dump` 272
Windows cluster
cluster failure during a restore 220
cluster group state during a restore 24, 220
disk requirements for 48
installing SnapManager into 47
maximum size supported by SnapManager 47
multiple-instance 47
system configuration requirements 47
Windows host system requirements
drive letters required for DBCC 136
Microsoft SQL Server 24
operating system 23
SnapDrive 24
SnapManager in workgroup mode 27, 41, 44
SnapManager licenses 25, 26
SnapManager service account 26
SQL Browser service 24
Windows operating systems with SnapManager
Windows 2000 Server
Microsoft hotfixes 23
SnapManager requirements 23
Windows Server 2003
Microsoft hotfixes 23
running a script from a UNC path on 347
SnapManager requirements 23
Windows XP theme 71
Windows XP theme 71
workgroup mode, Windows 27, 41, 44