SnapManager® 5.0 for Microsoft® SQL Server®
Installation and Administration Guide

NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089 U.S.A.
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 215-03557_A0
October 2008

Copyright and trademark information

Copyright information

Trademark information

SAN; SnapCache; SnapDirector; SnapFilter; SnapMigrator; SnapSuite; SohoFiler; SpinMirror; SpinRestore; SpinShot; SpinStor; vFiler; VFM Virtual File Manager; VPolicy; and Web Filer are trademarks of NetApp, Inc. in the U.S.A. and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of NetApp, Inc. in the U.S.A.

IBM, the IBM logo, AIX, and System Storage are trademarks and/or registered trademarks of International Business Machines Corporation.

Apple is a registered trademark and QuickTime is a trademark of Apple, Inc. in the U.S.A. and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the U.S.A. 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 U.S.A. and/or other countries.

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

NetApp, Inc. is a licensee of the CompactFlash and CF Logo trademarks. NetApp, Inc. NetCache is certified RealSystem compatible.

Table of Contents

Preface

About this guide

This guide describes how to install SnapManager® 5.0 for Microsoft® SQL Server® software. The guide also explains how to migrate, backup, restore, archive (at local and remote locations), clone, and recover database using SnapManager.

This guide describes how you can perform data management, data archival for long-term or remote storage of backups, and data replication for disaster recovery operations. It explains all the functionalities of SnapManager for Microsoft SQL Server. The appendixes provide details about the PowerShell command-line interface, the configuration wizard, and managing fractional space reservations.

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 database and system administrators who possess a working knowledge of SQL Server 2000, SQL Server 2005, or SQL Server 2008, and Windows® Server 2003, or Windows Server 2008. The discussion assumes familiarity with the following topics:

- SQL Server 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

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. Formulae and calculations.
`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

About this chapter

This chapter explains how SnapManager 5.0 for Microsoft SQL Server works and how it integrates with the rest of your environment. The chapter covers the following topics:

Related chapters:

- "Preparing to Install or Upgrade SnapManager" on page 21
- "Installing or Upgrading SnapManager" on page 43
- "Starting SnapManager for the first time after installation" on page 79

SnapManager overview

Contents of this section

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

New functionality with SnapManager 5.0

SnapManager 5.0 for Microsoft SQL Server supports the following new features:

- Control-file based configuration
- Integrity verification on SnapMirror® destination volume and SnapVault® secondary storage system
- SnapManager PowerShell cmdlet command-line functionality
- Database cloning wizard
- Dataset and SnapVault integration
- Transaction log backup compression (SQL Server 2008 specific feature)
- Filestream data type (SQL Server 2008 specific feature)
- Support for SQL Server 2008
- Support for Windows® 2008 IA-64 (Enterprise Edition)
- Support for SnapDrive 6.0.1
- Support for Microsoft® Management Console (MMC) 3.0

Control-file-based configuration: Control-file-based configuration enables you to perform all the operations that are executed currently through the Configuration wizard. Control-file based configuration is very useful in large configurations, disaster recovery, and mass deployment. Select the control-file-based configuration option through the Configuration wizard.

For more information, see “Understanding control-file based configuration” on page 127.:

Integrity verification on SnapMirror destination volume: SnapManager enables you to verify the SQL Server databases that are stored on the LUNs of the destination SnapMirror volumes. When verifying the integrity of a destination volume, SnapManager automatically detects the existing SnapMirror relationships in the SQL Server volumes and selects the available SnapMirror relationship for the selected destination volume.

For more information, see“Integrity verification on the SnapMirror destination volume” on page 280.

SnapManager PowerShell cmdlet command-line functionality: This feature enables you to execute operations using the new SnapManager command-line functionality. This new function allows you to create scripts to run SnapManager functionality without using the SnapManager graphical user interface (GUI).

Database cloning wizard: Database cloning is the process of creating a point-in-time copy of a production database or its backup set. Cloned databases can be used for multiple purposes:

- During application development cycles for testing functionality, implementing the functionality using the current database structure and content
- By data extraction and manipulation tools for populating data warehouses
- Recovering data that was mistakenly deleted or changed

SnapManager contains a clone wizard that provides a convenient interface for performing the following cloning operations:

- Cloning a database that is in production
- Cloning a database in a backup set
- Deleting cloned databases

Dataset and SnapVault integration: A dataset is a collection of storage sets that have identical data protection requirements on the primary storage. It is a component of Protection Manager and gives you extensive automated remote backup and archival capabilities with the help of SnapVault. A dataset includes the backups and replica of the primary database. It also includes configuration information for the primary database along with data protection policies that determine how the database is protected.

For more information, see “Dataset and SnapVault Integration” on page 343.

Transaction log backup compression: SQL Server 2008 introduces Database Mirroring Log Compression, in which the outgoing log stream from the source to the destination is compressed, minimizing the network bandwidth that database mirroring uses.

SnapManager supports transaction log backup compression. For more information, see the relevant SQL Server documentation.

Filestream data type: Unstructured data, such as text documents, images, and videos is often stored outside the database, separate from its structured data and outside transactional control.

Filestream data type adds transactional control to unstructured data stored in an NTFS file system. For more information, see the relevant SQL Server documentation.

Support for SnapDrive 6.0.1: SnapManager 5.0 for SQL Server is compatible with SnapDrive 6.0.1. SnapDrive versions before 6.0.1 are not supported with SnapManager 5.0.

What SnapManager does

SnapManager provides rapid online backup and near-instantaneous restoration of databases by using online Snapshot™ technology that is part of the Data ONTAP® software. SnapManager can also leverage the SnapMirror capabilities of storage systems to provide onsite or offsite SnapManager backup set mirroring for disaster recovery.

Data management: SnapManager supports the following data management capabilities:

- Migrating databases and transaction logs to LUNs on storage systems
- Backing up databases and transaction logs from LUNs on storage systems
- Verifying the backed-up databases and transaction logs
- Managing the SnapManager backup sets
- Restoring databases and transaction logs from previously created SnapManager backup sets

Data archiving for long-term or remote storage of backups: You can use SnapManager to create offline archives of Snapshot copies to unmanaged media for long-term retention. Three different archive methods are supported:

- Manually initiated archival using Network Data Management Protocol (NDMP) 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 archival of database backup sets using SnapManager is described in “Archiving SnapManager Backups” on page 325 and in “Performing Disaster Recovery with SnapManager” on page 285.

You can also create and restore database from archives at a remote location through dataset and SnapVault integration to SnapManager.

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 “Performing Disaster Recovery with SnapManager” on page 285.

Dataset and SnapVault integration: SnapManager helps you create, restore, and manage remote backups. Dataset and SnapVault technologies together form the basis of this integration.

For more information, see “Dataset and SnapVault Integration” on page 343.

What SnapManager does not do

SnapManager 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 earlier than SnapDrive 6.0.1.
- 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 does not support differential backup.
- SnapManager is not capable of backing up or restoring SQL Server databases that are accessed through CIFS.
- SnapManager does not backup the tempdb or other system database files.

Where you install and run SnapManager

You must install and run SnapManager on all SQL Server computers executing SnapManager operations.

Local administration: When you run SnapManager on the computer hosting SQL Server, it is called *SnapManager local administration*. System requirements for a SnapManager local administration are described in "Windows host system requirements" on page 27.

Remote administration: When you run SnapManager on a computer that is not hosting SQL Server, it is called *SnapManager remote administration*. If you install SnapManager on a computer different from the SQL Server computer, you can run SnapManager remotely to perform any task that you can perform on a locally installed SnapManager system.

System requirements for a SnapManager remote administration system are described in "Requirements for a remote administration server" on page 37.

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 local and remote administration system used for remote verification are described in "Requirements for a remote administration server" on page 37.

The setup and use of a remote verification server is described in Chapter 9, "Backing Up Databases Using SnapManager," on page 173.

About the SnapManager graphical user interface (GUI)

The SnapManager for SQL Server (referred to as SnapManager in the guide) user interface is a stand-alone graphical user interface based on the Microsoft Management Console 3.0 snap-in framework. The SnapManager GUI enables you to perform all the operations offered by SnapManager.

The new GUI enables you to:

- Manage and administer multiple instances of SnapManager successfully.
- Manage backup and restore operations of database files and transaction log files on LUNs.
- Schedule backups and verify the integrity of databases in SnapManager backup sets.
- Administer SnapManager on another server computer on the network.

- Configure database, transaction logs, and Simple Mail Transfer Protocol (SMTP) queue locations that are required for SnapManager backup and restore operations.

The user interface also enables you to schedule and automate backups and verify the integrity of databases in SnapManager backup sets.

SnapManager user interface includes the following components:

- Configuration Wizard including export and import
- Configuration Wizard Option Settings
- Backup Wizard
- Backup Settings
- Backup Verification Settings
- Clone Wizard
- Run Command After Operation
- Delete Backup
- Restore Wizard
- Fractional Space Reservation Settings
- Notification Settings
- License Settings
- Reconnect Server
- Disconnect Server
- Restore Settings
- View
- Refresh
- Help

Operations performed through the SnapManager command-line interface

SnapManager 5.0 supports the new SnapManager command-line functionality called *cmdlet*, through SnapManager PowerShell. This SnapManager command-line interface enables you to execute the following operations:

- new-backup
- verify-backup
- restore-backup
- get-backup
- delete-backup
- clone-database

- clone-backup
- delete-clone
- import-config
- export-config

For more information, see “SnapManager Command-Line Reference” on page 365.

How you use SnapManager

You can run SnapManager on your SQL server or on a different computer. When you run SnapManager on a different computer, it is called “SnapManager remote administration.” Using a SnapManager remote administration system, you can perform all of the tasks that you perform on a locally installed SnapManager system. When you perform database verification on a remote system, it is referred to as remote verification.

The following steps describe a typical way to use SnapManager:

- After installing SnapManager, you use the SnapManager Configuration wizard to migrate the database to a storage system. This involves dismounting your databases and moving them to LUNs on a storage system. The Configuration wizard ensures that your databases are placed correctly.
- After you configure data storage, you can use SnapManager Backup to create backups of the databases.
- If the need arises, you can use SnapManager Restore to restore your data (either entire groups of databases or individual databases) from one of the backups.

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

Terms and technologies

Contents of this section

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

A backup set consists of metadata located in the SnapInfo directory structure and Snapshot copies. The Snapshot copies are created in volumes containing LUNs used by databases that are contained in the backup set.

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

A database is a collection of logical objects within a physical structure. The physical structure consists of one or more data files, and one or more transaction log files. A database is either used by the SQL Server itself (system database) or by an application (user database).

Database Consistency Checker (DBCC)

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

host system

A computer that accesses storage on a storage system.

log shipping

A process that takes backed-up transaction logs from a primary SQL Server and applies them sequentially on a scheduled basis to another SQL Server database. If a failure occurs, an application could be redirected to the other server, which would be only slightly behind the primary database. Log shipping is a means of protecting organizations if a logical or physical system failure occurs.

MSCS

Microsoft Cluster Services (MSCS) are system services that make it possible to create a virtual system consisting of multiple cluster nodes; each node is an independent physical computer and is a failover resource of other nodes in the cluster. Each node can support one or more virtual SQL Server instances.

multiple-instance cluster

A multinode cluster with multiple virtual SQL Server instances. Each node can be active, running one or more virtual SQL Server instances or passive. The passive node is an idle system waiting for another node to fail over and thereby becoming an active node. If one system fails, the other system takes over its application services. See also “single-instance cluster” on page 12.

quorum disk

A shared disk resource that is 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. The quorum disk is a single-point-of-failure.

recovery model

There are three distinct ways that you can recover your SQL Server databases if a failure occurs. 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: It only supports full backup and not transaction log backup. Hence, it is not possible to perform point-in-time restore once the full backup is created, and only up-to-the-minute restore is possible.

Full: All transactions are logged.

Bulk logged: Certain database operations (including SELECT INTO, BULK COPY/BCP, CREATE INDEX, WRITETEXT, and UPDATETEXT) are logged minimally. Database pages changed by committed bulk-logged operation are copied to the backed-up transaction log. The Bulk logged model has a higher risk of data loss than 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 11.

storage system

Any NetApp storage product that supports Fiber Channel Protocol (FCP) 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

Structured Query Language.

SQL Server

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

SQL Server computer

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

SQL Server replication

A process that is initiated and controlled by the database engine (SQL Server).

system database

A type of database that is used internally by SQL Server. System databases are created either during installation or during feature configuration, such as the distribution database.

See also “user database” on page 14. 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. The database is created when replication is activated.

master database: Records the system-level information, SQL Server initialization information, and configuration settings for SQL Server. This database also records all login accounts and the mapping information from the name of a database to its primary file location.

tempdb database: A database that is used to fulfill all temporary storage needs, including stored procedures and tables. The tempdb database uses SQL Server during query processing and sorting, and for maintaining row versions used in Snapshot isolation. A clean copy of the tempdb database is recreated with 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 has 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 transaction log is a file that is used as a write ahead log. All transactional operations are recorded in the transaction log; a transaction is considered committed when the ‘commit’ transaction record has been written to the transaction log. The main purpose of the transaction log is for crash consistency; if there is a system crash, power failure, or similar disastrous event, then the transaction log has enough information to roll forward all committed transactions and roll back all noncommittal transactions.

user database

A database created for and used by an application is considered to be a user database. See also “system database” on page 13.

How SnapManager for Microsoft SQL Server works

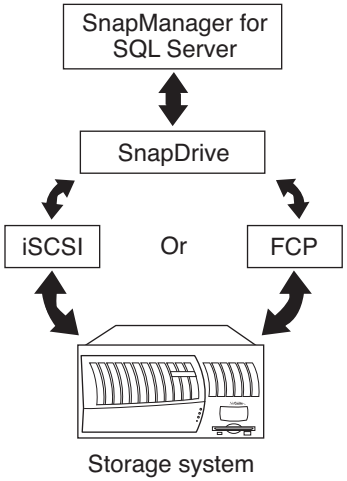
Contents of this section

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

System overview

SnapManager is a SQL Server aware application that provides backup and restore functionality in a SQL Server environment.

Relationship with other components of a SQL Server installation backed by a storage system: The following illustration shows the relationship between storage systems, SnapDrive, and SnapManager for Microsoft SQL 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 SnapDrive: You can use SnapDrive only to create, connect, expand, and manage LUNs.

SnapDrive manages LUNs on a storage system, making these disks appear as 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. For information about how to perform these tasks using SnapDrive, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

When to use SnapManager: Use SnapManager to migrate SQL Server databases from a local disk to a LUN and perform all operations on the databases.

Having SnapDrive installed on the SQL Server is a requirement for using SnapManager.

SnapManager and Snapshot copies

About SnapManager and Snapshot copies: A Snapshot copy is a point-in-time, read-only copy of a LUN stored on a storage system volume. SnapManager Backup uses Snapshot functionality to create real-time, online, read-only copies of databases. A SnapManager backup can consist of several Snapshot copies, depending on how your data is configured.

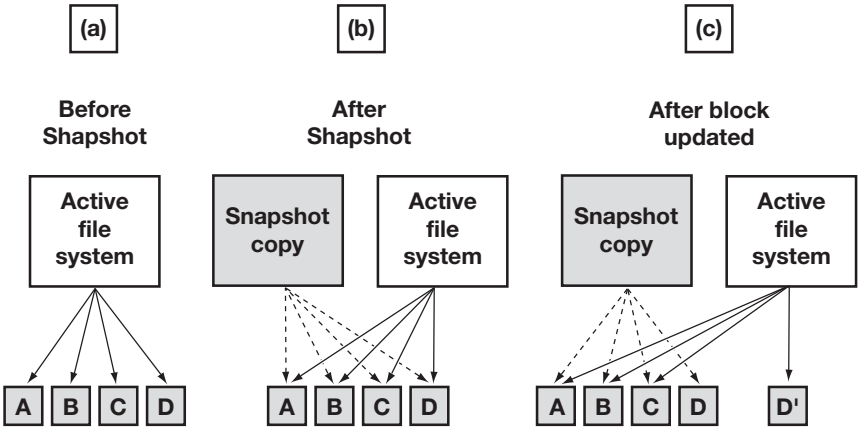
Note

Always use SnapManager to manage SnapManager backups, rather than managing the backup sets using SnapDrive or storage system administration tools, such as the FilerView® interface.

When to use the various Snapshot copy and backup methods: There are multiple ways to take Snapshot copies or backups in an installation that includes SnapManager. It is important to understand when each of these methods can produce a restorable image and when they cannot.

When one or more databases are mounted, Snapshot copy-based backups should be performed using only SnapManager. Creating Snapshot copies using the storage system console or FilerView results in an inconsistent HTFS file system hosted by the LUNs in the Snapshot copy. Using SnapDrive to create Snapshot copies creates inconsistent database images.

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



Example: 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 the disk store both versions, so the Snapshot copy version of file.txt consumes no more disk space.

Now, 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 refer to the three original disk blocks that have not been modified since the Snapshot copy, plus the one new block. The Snapshot copy inodes still refer to the original four blocks.

If you delete file.txt, the blocks holding its data are no longer part of the active file system.The blocks still remain a 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.

Maximum number of Snapshot copies you can retain: Data ONTAP software allows a maximum of 255 Snapshot copies per storage system volume. Because SnapManager backups create Snapshot copies, you must delete old SnapManager backups because they are no longer needed. Ensure you delete older backups to avoid reaching the limit of 255 Snapshot copies per storage system volume.

Note

The number of Snapshot copies on a volume can be greater than the number of SnapManager backups being retained. For example, if a single volume contains both the SnapInfo directory and the databases, each SnapManager backup generates two Snapshot copies on that volume.

How SnapManager works with other backup methods

It is recommended 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 recovery process, you should not perform transaction log backups with any application other than SnapManager.

What SnapManager does not back up: SnapManager does not backup all the files commonly used by a SQL Server computer. You can use NTBackup (a native backup utility that ships with Windows) to backup 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 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

About this chapter

This chapter describes tasks you need to perform before installing or upgrading to SnapManager 5.0. The following topics are covered in this chapter:

Note

For the most current list of system requirements, see the *SnapManager 5.0 for Microsoft SQL Server Product Description page* on the NOW™ (NetApp on the Web)® site at http://now.netapp.com/NOW/cgi-bin/software/

Note

For information about compatible versions of SnapManager, SnapDrive, and Data ONTAP, see the *NetApp Interoperability Matrix* on the NOW™ site at http://now.netapp.com/NOW/products/interoperability/

Preinstall or preupgrade procedure

Prerequisites for installing or upgrading SnapManager

Before you begin installing or upgrading SnapManager, you must complete the following tasks.

Task	Process
1	Back up system resources and databases, as described in "Backing up system resources and data" on page 25.
2	Determine whether you want to use per-SQL-server SnapManager licensing or per-storage-system SnapManager licensing. For more information, see "Verifying Windows host system requirements" on page 26.
3	Configure or upgrade your storage system according to the requirements for SnapManager and SnapDrive, described in "Verifying storage system requirements" on page 39. You might need to install or upgrade any of the following components: ◆ Data ONTAP ◆ iSCSI and/or FCP protocols ◆ SnapManager license ◆ SnapRestore® license ◆ SnapMirror license ◆ FlexClone® license ◆ SnapVault license (for primary and secondary locations) **Note** You need SnapVault licenses only if you have Protection Manager installed on your system for dataset and SnapVault integration to SnapManager. For more information, see the relevant Protection Manager documentation.

<table>
<tr><th>Task</th><th colspan="2">Process</th></tr>
<tr><td rowspan="3">4</td><td>If...</td><td>Then...</td></tr>
<tr><td>You upgrade SnapManager and you also upgrade underlying SnapDrive or Microsoft iSCSI initiator versions</td><td>Make a note of this now. Later, while preparing to upgrade the SnapManager application (described in“Uninstalling SnapManager” on page 70), you must remove the iSCSI dependency with respect to SnapManager.</td></tr>
<tr><td>You upgrade only SnapManager</td><td>Go to step 5.</td></tr>
<tr><td>5</td><td colspan="2">Note whether your storage system has multiple IP addresses.</td></tr>
<tr><td>6</td><td colspan="2">Configure or upgrade your Windows host systems to meet the requirements for SnapDrive and SnapManager, described in “Verifying Windows host system requirements” on page 26.

You might need to install, upgrade, or configure the following components:
◆ Microsoft Windows operating system
◆ Microsoft Windows hotfixes
◆ SnapDrive
◆ SnapDrive preferred IP address (if your storage system has multiple IP addresses)
For details, see the SnapDrive Installation and Administration Guide for your version of SnapDrive.

Note
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 LUN on a storage system.

◆ iSCSI and/or FCP protocols</td></tr>
<tr><td>7</td><td colspan="2">Be sure that the TCP port 808 is open for SnapManager to function.</td></tr>
</table>

Task	Process
8	After you complete these tasks, you are ready to install or upgrade SnapManager. Go to “Installing or Upgrading SnapManager” on page 43.

Backing up system resources and data

Backing up system resources and data

Before you install SnapManager, you are strongly advised to backup your system resources and data that uses Windows NT Backup or another industry standard backup utility.

To backup your system resources and data, complete the following steps.

Step	Action
1	Back up the operating system installation on the SQL server, including the system state.
2	Back up the data on the local drives on the SQL server.
3	Back up the boot and system drives, and the registry.
4	Use your backup utility to create and maintain a current emergency repair disk (ERD).

Verifying Windows host system requirements

Contents of this section

In the most basic configuration, SnapManager is installed on the same Windows host system as SQL Server. 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 the previously listed configurations of SnapManager:

- "Windows host system requirements" on page 27
- "Windows operating system requirements" on page 31

Note

For the most current information about the Windows host system requirements, see the *SnapManager 5.0 for Microsoft SQL Server Description Page* at the NOW site.

Windows host system requirements

The following table lists the Windows host system requirements:

Windows host component	Requirements
Operating system	One of the following for SnapManager 5.0 for the SQL Server (x86) configuration: ◆ Windows Server 2003 Standard Edition with SP2 (x86) ◆ Windows Server 2003 Enterprise Edition with SP2 (x86) ◆ Windows Server 2003 R2 Standard Edition with SP2 (x86) ◆ Windows Server 2003 R2 Enterprise Edition with SP2 (x86) ◆ Windows Server 2008 Standard Edition (x86) ◆ Windows Server 2008 Enterprise Edition (x86) One of the following for SnapManager 5.0 for the SQL Server (x64) configuration: ◆ Windows Server 2003 Standard Edition (SP2, x64) ◆ Windows Server 2003 Enterprise Edition (SP2, x64) ◆ Windows Server 2003 R2 Standard Edition (SP2, x64) ◆ Windows Server 2003 R2 Enterprise Edition (SP2, x64) ◆ Windows Server 2008 Standard Edition (x64) ◆ Windows Server 2008 Enterprise Edition (x64) For the SnapManager 5.0 for SQL Server (IA-64) configuration: ◆ Windows Server 2008 Enterprise Edition (IA-64)
Microsoft Windows hotfixes	See the SnapDrive software system requirements.

Windows host component	Requirements
Microsoft SQL Server	One of the following for SnapManager 5.0 for the SQL Server (x86) configuration: ◆ SQL Server 2005 Standard Edition with SP2 (x86) ◆ SQL Server 2005 Enterprise Edition with SP2 (x86) ◆ SQL Server 2008 Standard Edition (x86) ◆ SQL Server 2008 Enterprise Edition (x86) ◆ SQL Server 2000 Standard Edition with SP4 (x86) ◆ SQL Server 2000 Enterprise Edition with SP4 (x86) ◆ SQL Server 2008 Standard Edition (x86) ◆ SQL Server 2008 Enterprise Edition (x86) One of the following for SnapManager 5.0 for the SQL Server (x64) configuration: ◆ SQL Server 2005 Standard Edition (SP2, x64) ◆ SQL Server 2005 Enterprise Edition (SP2, x64) ◆ SQL Server 2008 Standard Edition (x64) ◆ SQL Server 2008 Enterprise Edition (x64) One of the following for SnapManager 5.0 for the SQL Server (IA-64) configuration: ◆ SQL Server 2005 Enterprise Edition with SP2 (IA-64) ◆ SQL Server 2008 Enterprise Edition (IA-64) **Note** The preceding requirement does not apply to a remote administration server.

Windows host component	Requirements
SQL Server Browser service	If the host system is running SQL Server 2005, configure the SQL Browser service to start automatically.
SnapDrive	SnapDrive 6.0.1 and later If you need to install or upgrade SnapDrive, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive for detailed instructions. **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.
SnapDrive preferred IP address	If your storage system has multiple IP addresses, configure the SnapDrive preferred IP address. 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 LUN on a storage system.
SnapManager Licenses	If SnapManager is licensed per-server, a SnapManager license is required on the Windows host system. **Note** For per-server SnapManager licensing, you can install SnapManager without specifying a server-side license; after SnapManager has been installed, you can apply the license from the License Settings dialog box.

Windows host component	Requirements
Microsoft .NET Framework 3.5	The SnapManager installation package installs Microsoft .NET Framework 3.5 if it is not present in the host system.
Microsoft Management Console (MMC)	You require the MMC 3.0 x64 Edition or MMC 3.0 x86 Edition to launch the SnapManager snap-in console for starting the SnapManager installation.
Windows PowerShell	PowerShell 1.0 RTM x86 Edition or PowerShell 1.0 RTM x64 Edition. This is a prerequisite before you run the SnapManager installation.
Other hardware and software	See your SnapDrive documentation for complete details about the following system requirements: ◆ Host hardware operating system ◆ LUN access protocol (FCP or iSCSI) software The preceding requirements do not apply to a remote administration server.

Before installation, keep in mind the following points:

- If your system has SQL Server 2008 installed, then during the installation of SnapManager 5.0, SnapManager automatically installs the required SQL Server 2005 backward compatibility components (if they are not already installed). If you install SQL Server 2008 after upgrading or installing SnapManager, install the SQL Server 2005 backward compatibility components manually.
- If the host system is running SQL Server 2005, Microsoft Data Access Components (MDAC) 2.8 SP1 must be installed. Windows Server 2003 SP1 and SP2 include MDAC 2.8 SP2, which is required for SQL Server 2005 on Windows Server 2003.
- Be sure that the TCP port 808 is open for SnapManager to function.
- Fractional space reservation is disabled by default. If you are not using fractional space reservation on the storage system volumes that contain LUNs that are used for SQL Server, the monitoring can be disabled. Doing so improves backup completion times. For details on how to enable and

disable monitoring, see “Managing fractional space reservation” on page 455.

Windows operating system requirements

The following table gives the Windows operating system requirements.

Operating system	Requirements
Windows 2003 Server	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
Windows 2008 Server	◆ Microsoft Windows Server 2008 Standard Edition (x64) ◆ Microsoft Windows Server 2008 Standard Edition (x86) ◆ Microsoft Windows Server 2008 Enterprise Edition (x64) ◆ Microsoft Windows Server 2008 Enterprise Edition (x86) ◆ Microsoft Windows Server 2008 Enterprise Edition (IA-64)

Note

SnapManager offers support only to Windows guest operating systems of VMWare® ESX 3.0.2 (Update 1), VMware ESX 3.5 (with updates), and the Windows child operating system of Microsoft Hyper-V™.

Preparing a Windows host system for SnapManager installation

Before you install SnapManager

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. 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 *locally* from this host, then go directly to Step 2. ◆ If you plan to administer SnapManager *remotely* from this host, review "Requirements for a remote administration server" on page 37 to determine your installation requirements. You do not need to install SnapDrive unless you want to use the remote administration server to remotely administer SnapDrive or you want to use the remote server for verifying backups. To verify backups from a remote server, you should mount the Snapshot copy to a LUN through SnapDrive. ◆ If you plan to use SnapManager on this host only to perform *remote verification*, then review "Requirements for a remote verification server" on page 37 to determine your installation requirements.

Step	Action
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 according to the instructions in "Where you install and run SnapManager" on page 7.

Note

Be sure that the TCP port 808 is open for SnapManager to function.

SnapManager license requirements

SnapManager license requirements

SnapManager 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.

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). You can also 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-storage system license or a storage system-side license. If no server-side license is detected, SnapManager checks the storage system for a storage system-side license when a SnapManager operation is initiated. If a storage system-side license is not enabled, the SnapManager operation fails and an error message is written to the Windows event log.

SnapManager service account requirements: 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.

Authentication

SQL Server authentication

When using SQL Server authentication, the SQL Server administrator must have sysadmin server role privileges on the SQL Server instance. SnapManager requires that the database administrator have the required privileges to mount and unmount databases and backup and restore data and transaction log 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.

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 24. 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: To use SnapManager to migrate, backup and restore SQL Server data, you must have SnapDrive installed on the same computer and use it to create two or more dedicated LUNs to hold the SQL Server databases and SnapInfo. SnapInfo cannot share a LUN with any database files.

Note

It is advisable that you create a minimum of three LUNs so that 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 is used to administer the Microsoft SQL Server computer that is running with SnapManager. If this remote system is not 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 to offload 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. The system requirements for a storage system used in a SnapManager environment are described in "Verifying storage system requirements" on page 39.

Remote servers

Requirements for a remote administration server

A SQL Server instance 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 420.

You can add servers that you want to use through the option "Add servers to be managed" in the Actions pane. For more information, see "Add new servers" on page 90.

Requirements for a remote verification server

SnapManager performs remote verification 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, in addition to 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.

Note

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.

- The remote Windows system must have connectivity (iSCSI or FCP) to the storage system.

  Note

  If you are using iSCSI to connect to the storage system on the remote verification server, an iSCSI connection must be created.

- The SnapManager version and SnapDrive version on both the remote computer and host computer should be same.
- Microsoft SQL Server 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.

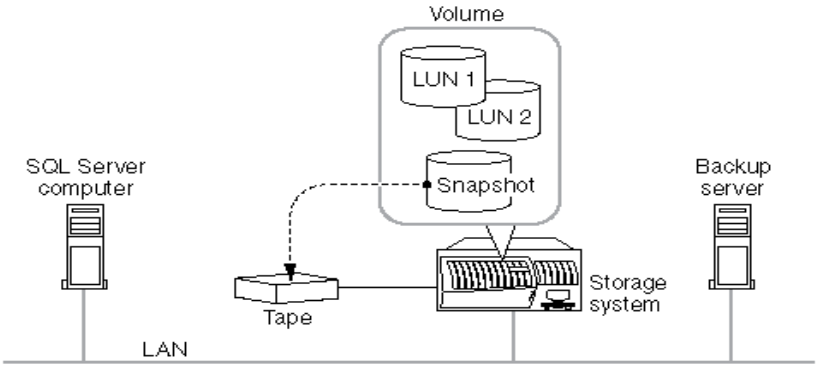
Note

If you cannot use a remote SQL Server instance or a local non-clustered 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.

Note

Although it is possible to restore from an unverified backup, you should restore only from verified backups.

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



Archiving using NDMP or dump

Verifying storage system requirements

Contents of this section

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

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.
iSCSI protocol or FCP	The appropriate LUN access protocol software must be installed on the storage system that stores the databases. For more information, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.
FlexClone® license	A FlexClone license is required. **Note** If you do not have FlexClone licensed, SnapManager does not support the verification on destination SnapMirror volumes. Also, the cloned databases will be unable to use FlexClone.

Storage system component	Requirements
SnapMirror license	If you plan to use the SnapMirror software along 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.
Additional requirements	All the SnapDrive requirements that apply: ◆ SnapRestore feature ◆ iSCSI and/or FCP protocols ◆ SnapMirror license (if in use) ◆ SnapVault licenses on the primary and the secondary storage systems ◆ Protection Manager 3.7 D8 license ◆ Preferred IP address (if multiple IP addresses)

About storage systems that have multiple IP addresses

If your storage system has multiple IP addresses, configure the SnapDrive preferred IP address. 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 LUN on a storage system.

Additional SnapDrive 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 along 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 more information, 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.

Installing or Upgrading SnapManager 3

About this chapter

The following topics describe the procedures for installing or upgrading to SnapManager 5.0 for SQL Server and then starting the application for the first time. Other sections pertain to tasks you might need to use at some point after SnapManager has been installed.

Installing SnapManager 5.0 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 44
- "System configurations for SnapManager on a Windows cluster" on page 55

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

- "Upgrading to SnapManager 5.0" on page 62

After you have installed SnapManager 5.0, start the application for the first time as described in Chapter 4, "Starting SnapManager for the first time after installation," on page 79.

After SnapManager 5.0 for SQL Server has been installed

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

- "Uninstalling SnapManager" on page 70

  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 77

Installing SnapManager on a stand-alone Windows host system

About this section

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:

- "Modes of installing SnapManager" on page 44
- "Installing in interactive mode" on page 46
- "Installing SnapManager in unattended installation mode" on page 49
- "System configurations for SnapManager on a Windows cluster" on page 55

Note

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

Modes of installing SnapManager

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	Requires user interaction and access to the user interface	Allows automated installation by executing a script or command line.
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 select 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.

Installing in interactive mode

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

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">Install the software from the CD that came packaged with your media kit 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</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>2</td><td colspan="2">In the Welcome screen, click Next.

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

Step	Action
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.
4	In the Customer Information screen, specify the user name, the organization name, and the SnapManager license type. See “SnapManager license requirements” on page 34 for details about the two license types. If you have a storage-system-side license for SnapManager, select Per storage system, 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 and use the License Key box to enter the license key for your server-side license. **Note** 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 by selecting Help > About the SnapManager interface. From that dialog box, you 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.
6	In the destination folder, note the full path of the folder in which SnapManager will be installed. The default installation directory for SnapManager 5.0 for Microsoft SQL Server is as follows: `c:\Program Files\NetApp\SnapManager for SQL Server\`

Step	Action
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. Record the newly specified installation directory path.
8	Click Next. **Result:** The SnapManager Server Identity screen appears.
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. If SnapDrive is installed and configured, the text box is populated with the account for which SnapDrive is configured. Otherwise, browse to find 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 the installation specifications are complete.

Step	Action
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.
14	After the InstallShield Wizard Completed screen appears click Finish to exit the software installation utility.
15	Proceed to Chapter 4, "Starting SnapManager for the first time after installation," on page 79.

Installing SnapManager in 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 "Installing SnapManager in unattended installation mode" on page 49. For a comparison of the two installation modes, see "Modes of installing SnapManager" on page 44.

How to run the software installation utility: The command name you specify depends on the SnapManager installation media you access.

SnapManager installation media	Application to be run
Software CD	`setup.exe` (located on the CD)
Software package downloaded from the network	`SMSQL5.0.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 `SMSQL5.0.exe`, depending on the SnapManager installation media 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 `SMSQL5.0.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 does not proceed unless you specify this parameter with a value of `Yes`. If not specified, the default value `No` is used, and the installation does not proceed.

Command or parameter	Description
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 a SQL server-side license for SnapManager.
INSTALLDIR=*InstallationDirectory*	Optional. If not specified, the default installation directory is used: `C:\Program Files\NetApp \SnapManager for SQL Server\`
SVCUSERNAME=*Domain\UserName*	The account from which SnapManager is to be run.
SVCUSERPASSWORD=*Password*	
SVCCONFIRMUSERPASSWORD=*Password*	
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 is not be rebooted.

Command or parameter	Description
`/L* TempDirPath\LogFileName`	Optional. If you specify this option, detailed information about the installation is 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 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 is created or overwritten. *LogFileName* is the name of the file to which the transaction logs are written.

Examples: Consider the following two scenarios.

Installation details	Scenario 1	Scenario 2
Installation media to be used	CD inserted into the CD 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	Storage system-side	Server-side (license key `ABCDEFGHIJKLMN`)

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

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

or

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

For scenario 2, replace `setup.exe` with the downloaded binary name in the command given above.

```
C:\NetApp\downloads\SMSQL5.0.exe /s /v"AGREETOLICENSE=Yes
SVCUSERNAME=MKTG2\Administrator SVCUSERPASSWORD=****
SVCCONFIRMUSERPASSWORD=***  /qb"
```

or

```
C:\NetApp\downloads\SMSQL5.0.exe/s /v"AGREETOLICENSE=Yes
SVCUSERNAME=MKTG2\Administrator SVCUSERPASSWORD=****
SVCCONFIRMUSERPASSWORD=*** ISX_SERIALNUM=*** /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\`.

Step	Action
3	Proceed to Chapter 4, "Starting SnapManager for the first time after installation," on page 79.

System configurations for SnapManager on a Windows cluster

System configurations for SnapManager on a Windows cluster

When installing SnapManager on an MSCS cluster, consider the following system configuration requirements and limitations. See "Volume mount point limitations in a clustered environment" on page 105 for the different cluster models supported.

Note

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

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 to eight nodes, depending on the operating system.

Operating system	Maximum cluster size
Windows Server 2005 Enterprise Edition	Four nodes
Windows Server 2008 Enterprise Edition	Eight 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 specified instance must be in the cluster group for that instance and in the SQL Server list of dependencies.

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, if your SQL Server data is on a shared resource, your disk requirements are the same as for a stand-alone SQL Server system. A LUN gets added for the quorum disk. A minimum of three LUNs are required:

- One LUN for the databases
- One LUN for the SnapInfo directory
- One LUN if a shared quorum disk is used

Multiple-instance cluster example: In an Active/Active two-node configuration, there are two clustered nodes and a 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, if a shared quorum disk is used. 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 databases
  - One LUN to store the SnapInfo directory
- For node 2
  - One LUN to store the SQL Server 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 time.

For more information about MSCS clustering with SQL Server, see the *SQL Server 2005 Failover Clustering* document at the following URL: http://www.microsoft.com/downloads/details.aspx?FamilyID=818234dc-a17b-4f09-b282-c6830fead499&displaylang=en.

Installing SnapManager and creating a new Windows cluster

Installing SnapManager and creating a new Windows cluster

The procedure for installing and creating a new Windows cluster assumes the following:

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

Note

Install MMC 3.0 and Windows PowerShell 1.0 (x86, x64 or IA-64). If Microsoft .NET Framework 3.5 is not already installed, SnapManager automatically installs it.

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 Server 2003 Standard Edition with Service Pack 2 ◆ Windows Server 2003 Enterprise Edition with Service Pack 2 ◆ Windows Server 2003 Standard Edition R2 with Service Pack 2 ◆ Windows Server 2003 Enterprise Edition R2 with Service Pack 2 ◆ Windows Server 2008 Standard Edition Windows Server 2008 Enterprise Edition See your Microsoft documentation for information about 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.

Step	Action
3	Create a quorum disk, if a shared quorum disk is used. 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 are 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.
6	Install SQL Virtual Server on the MSCS cluster. See your SQL Server documentation for information about how to do this.
7	If you are using SnapDrive, 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 Knowledgebase article KB10590 “Restoring a Clustered LUN Resource without Taking the LUN Resource Offline” on the NOW site the article “Restoring a Clustered LUN Resource without Taking the LUN Resource Offline with SnapDrive and SnapManager for SQL” on the NAS Service and Support site.

Step	Action
8	Install and configure SnapManager *on all nodes of the cluster.* To do this, go to "Installing SnapManager on a stand-alone Windows host system" on page 44, and use the procedures described in "Upgrading using the interactive mode" on page 63. **Note** You do not need to stop SQL Server instances before or during the SnapManager software upgrade process.
9	Start SnapManager (as described inChapter 4, "Starting SnapManager for the first time after installation," on page 79).
10	Run the Configuration wizard (as described inChapter 7, "Using the SnapManager Configuration Wizard," on page 121).
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), restart the SnapManager user interface and run SnapManager Backup.

Installing SnapManager in an existing Windows cluster

Installing SnapManager in an existing Windows cluster

To install and configure SnapManager 5.0 in an existing Windows cluster, complete the following steps.

Note

Before you complete these steps, ensure that you have prepared your system and environment as described in "Preinstall or preupgrade procedure" on page 22.

Step	Action
1	Verify that the virtual servers and the cluster services are functioning by moving the virtual server from one cluster node to the other and back. If any errors occur, or if any of the cluster resources do not start, resolve the issue before continuing.
2	Install or upgrade SnapDrive as required. For details, see your SnapDrive documentation.
3	From the node that owns the cluster group that contains the virtual server, create the shared LUNs to hold the databases and transaction log files. For details, see your SnapDrive documentation. **Note** These shared LUNs must be physical disk resources in the cluster group that contains the virtual server that uses them.

Step	Action
4	Verify that the System Attendant Resource dependencies are set correctly. **Note** SnapManager adds the dependencies automatically for all the LUNs that it uses, if the Configuration wizard detects that it is running on a cluster. **Note** There are no dependencies set on the LUN physical disk resource in a Continuous Cluster Replication configuration.
5	Verify that the virtual servers and the cluster services are functioning correctly by moving the cluster group containing the newly created virtual server to the other node and back.
6	Install SnapManager on all nodes, starting with the node that currently owns the cluster resources. **Note** You need to install MMC 3.0, Windows PowerShell 1.0 (x64, x86 or IA-64) and Microsoft .NET 3.5 Framework (x64, x86 or IA-64) on 64-bit and 32-bit operating systems, respectively, for SnapManager to work. Use either the interactive installation procedure or the unattended installation procedure for a stand-alone Windows host system. Both procedures are described in "Installing SnapManager on a stand-alone Windows host system" on page 44.
7	Go to "Installing or Upgrading SnapManager" on page 43.

Upgrading to SnapManager 5.0

About this section

This section contains procedures for upgrading to SnapManager 5.0 for Microsoft SQL Server on a Windows host system that is running SnapManager 5.0. See the following topics for more information:

- "Pre-upgrade checklist" on page 62
- "Comparison of the two upgrade modes" on page 63
- "Upgrading using the interactive mode" on page 63
- "Upgrading in unattended mode" on page 66

Note

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

Pre-upgrade checklist

Before you upgrade to SnapManager 5.0 for Microsoft SQL Server, be sure that your Windows host systems and the storage system are running the supported software versions, as described in "Preparing to Install or Upgrade SnapManager" on page 21.

The following paragraphs summarize these software upgrade considerations.

Microsoft SQL Server: 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 26.

SnapDrive: SnapManager 5.0 for Microsoft SQL Server is supported by SnapDrive 6.0.1. For more details, see "Verifying Windows host system requirements" on page 26.

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 39.

For the latest information: For the most current information about SnapManager and SnapDrive, see "Verifying Windows host system requirements" on page 26. The listed documents (and all other NetApp documents) are available at the NOW site at http://now.netapp.com/NOW/knowledge/docs.

Note

SnapManager has fractional space reserve monitoring enabled by default. When upgrading, fractional space reservation remains enabled. Otherwise, it is disabled by default. If you are not using fractional space reservation on the storage system volumes that contain LUNs that are used for SQL Server, the monitoring can be disabled. Doing so improves backup completion times.

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.

Feature	SnapManager upgrade mode	
	Interactive	**Unattended**
Access	Requires user interaction and access to the user interface	Allows automated upgrade by executing a script or command line
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

Upgrading using the interactive mode

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

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, backup 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. Use the NTBackup utility 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 must upgrade your version of SnapDrive. For details, see "Verifying Windows host system requirements" on page 26.
4	If you have not already done so, verify that your storage system meets the minimum requirements. **Note** If the Windows host system is running SnapManager 1.5 for Microsoft SQL Server, you must 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 39.

<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 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 SnapManager for SQL Server-InstallShield Wizard starts and opens to the Welcome screen.</td></tr>
<tr><td></td><td>You are installing the software from a CD</td><td>Browse to the SnapManager installation package and double-click <code>setup.exe</code>.

Result: The SnapManager for SQL Server-InstallShield Wizard 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 option selected and then click Next.

Result: The Ready to Repair the Program screen appears.</td></tr>
</table>

Step	Action
8	In the Ready to Repair the Program screen, click Install. **Result:** The installation begins. When the installation completes, the InstallShield Wizard Completed screen appears.
9	In the InstallShield Wizard Completed screen, click Finish to exit the software installation utility.
10	Go to Chapter 4, "Starting SnapManager for the first time after installation," on page 79.

Upgrading in unattended 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.

How to run the software installation utility: The command name you specify depends on the SnapManager installation media you access.

SnapManager installation media	Application to be run
Software CD	`setup.exe` (located on the CD)
Software package downloaded from the network	`SMSQL5.0.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 /s /v"[/L* TempDirPath\LogFileName] /qb"
```

The following table describes each of the parameters.

Command or Parameter	Description
CommandName	Either `setup.exe` or `SMSQL5.0.exe`, depending on the SnapManager installation media being used.

Command or Parameter	Description
/L TempDirPath\LogFileName*	Optional. If you specify this option, detailed information about the installation is 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 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 is created or overwritten. *LogFileName* is the name of the file to which the installation information is written.

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	SnapManager for SQL Server 5.0	

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

```
E:\setup.exe /s /v"/qb"
```

For scenario 2, replace `setup.exe` with the downloaded binary name in the command given above.

```
C:\NetApp\downloads\SMSQL5.0.exe /s /v"/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 after installation," on page 79.

Updating the scheduled legacy jobs using Legacy Job Upgrader

You can upgrade the scheduled jobs that you created using the Windows Task Scheduler or SQL Server Agent in the older versions of SnapManager. To update the SnapManager legacy scheduled jobs to SnapManager 5.0, complete the following steps.

Step	Action
1	Launch "SMSQLUpgradeJobs.exe" from the SnapManager Installation directory. **Result:** The "Update SnapManager for SQL Server legacy scheduled jobs" window appears with all the SnapManager legacy scheduled jobs listed for the selected server.

Step	Action
2	To see the jobs in a different server, use Browse to select a different server and click Refresh. **Result:** SnapManager lists the legacy scheduled jobs for the selected server.
3	You can select Windows Task Scheduler or SQL Server Agent by selecting the corresponding radio button.
4	To update the legacy scheduled jobs, click Update. **Result:** A Scheduling dialog box appears that you can use to migrate the legacy scheduled jobs to SnapManager 5.0.

In an MSCS cluster environment, Job Upgrader shows all the nodes in a list. You can select a specific node and migrate a legacy job to that particular node. It is recommended that you schedule the job on all nodes in the cluster, to achieve fault tolerance.

In earlier versions of SnapManager, the jobs that are scheduled to run against a server are not required to reside in the same server. In SnapManager 5.0, the jobs that are targeted to run against a server need to be scheduled in that particular server's scheduler.

By default, SnapManager enables the “Delete legacy job” and “Replace the job if it exists” check boxes, if the target server is different from the server on which the legacy scheduled jobs exist or if the name of the specified job is different from the legacy scheduled job.

Uninstalling SnapManager

About this section

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

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 44
- "System configurations for SnapManager on a Windows cluster" on page 55
- "Upgrading to SnapManager 5.0" on page 62.
- "Reinstalling SnapManager" on page 77

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.

Multiple 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 no longer has 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.

Feature	SnapManager uninstallation mode	
	Interactive	**Unattended**
Access	Require user interaction and access to the user interface. You can also uninstall using the .exe file that you used to install the software.	Allows automated uninstallation by executing a script or command line
Tool used	The Windows utility Add or Remove Programs (in 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 73. For a comparison of the two uninstallation modes, see "Comparison of the two upgrade modes" on page 63.

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 continues to run during the uninstallation process and after the uninstallation completes.	
2	In the Windows Server 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 on 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.** Select the Remove Reports option. **b.** 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 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 72. For a comparison of the two uninstallation modes, see "Comparison of the two upgrade modes" on page 63.

How to run the software installation utility: The command name you specify depends on the SnapManager installation media you access.

SnapManager installation media	Application to be run
Software CD	`setup.exe` (located on the CD)
Software package downloaded from the network	`SMSQL5.0.exe` (the name of the software package)

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 `SMSQL5.0.1.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 is 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 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 is created or overwritten. *LogFileName* is the name of the file to which the installation information is written.

Examples: Consider the following two scenarios.

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

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

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

For scenario 2, replace `setup.exe` with the downloaded binary name in the command given above.

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	Go to Chapter 4, "Starting SnapManager for the first time after installation," on page 79.

Note

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

Reinstalling SnapManager

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 44
- "System configurations for SnapManager on a Windows cluster" on page 55

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 before the reinstallation.

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

For more information, see "Uninstalling SnapManager" on page 70.

If you did not uninstall SnapManager: See "Upgrading to SnapManager 5.0" on page 62.

Migrating SnapManager to a new hardware

Migrating SnapManager to a new hardware

If you migrate the host Windows operating system that runs SnapDrive and SnapManager for SQL to a new hardware, follow these steps to reconnect to an SQL server database before or after the migration.

Step	Action
1	Detach all of the databases and note the database names.
2	Note the drive letters or mount points.
3	Unmount the LUNs from the server.
4	Perform a fresh install of the operating system on the new hardware. **Note** Avoid two different iSCSI hosts attempting connection to the same LUN that runs the SQL server database as it leads to database corruption.
5	Configure the new iSCSI initiator to point to the pre-existing LUNs.
6	Connect to the disks by configuring SnapDrive.
7	Run the SnapManager for SQL Configuration wizard.
8	Attach all of the databases when all of the LUNs are connected with the same drive letters or mount points they were connected to before the disconnection.

Starting SnapManager for the first time after installation 4

About this chapter

This chapter describes how to start SnapManager for the first time after you have installed SnapManager. It contains the following topics:

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 messages and dialog boxes at the system console.

How to start SnapManager for the first time after installation

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

Step	Action
1	From the Start menu, select Program Files > NetApp >SnapManager for SQL Server. If you have upgraded to SnapManager 5.0 for Microsoft SQL Server, SnapManager automatically connects to the default SQL Server already configured. Go to Step 4. If you have installed SnapManager for the first time, you can specify the default SQL Server using the 'Add Servers to be Managed' option. Go to Step 2.

Step	Action
Specify the default SQL Server.	
2	Select the SQL Server you want to manage. **a.** In the Actions pane, click 'Add Servers to be Managed'. Select a Server from the list, or use the Browse option to select a server, then click 'Add'. In a clustered configuration, you can add only a virtual server using the option "Add servers to be managed." **Note** You can also add server instances using this option, as there can be many server instances within a domain or a single physical server. **b.** In the login details box, select 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 information, see "Connecting to a SQL Server instance" on page 418. **Result:** SnapManager connects to the specified SQL Server, using the specified security authentication method.

Step	Action
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.
SnapManager checks the SQL Server version.	
4	If SnapManager detects that the SQL Server 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 27 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.** Select 'Add Servers to be Managed'. To do this, select Action > Add Servers to be Managed. **b.** Return to Step 2. ◆ Otherwise, close the SnapManager application, upgrade Microsoft SQL Server as needed, then restart SnapManager.

Step	Action
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 27 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.
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 27 and note the requirements for Microsoft SQL Server. 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. After you have planned your database migration, start the Configuration wizard manually and specify database and LUN configuration details (as described in Chapter 7, "Using the SnapManager Configuration Wizard," on page 121).

What to do next

Overview

Before you run the Configuration wizard, you must plan your database migration. Database migration planning is covered in the following chapters:

- "Configuration and volume mount points" on page 93
- "Using the SnapManager Configuration Wizard" on page 121

Note

The information in the section instructs you to start the Configuration wizard manually and then describes how to specify your database and LUN configuration details.

Understanding the SnapManager GUI 5

About this chapter

SnapManager 5.0 and preceding provides an enhanced, easy-to-use graphical user interface (GUI). The SnapManager GUI is based on Microsoft Management Console (MMC) 3.0.

This section describes the different SnapManager components that you need to use frequently when you work with SnapManager. The following topics are covered in this chapter:

SnapManager snap-in

The SnapManager snap-in is based on Microsoft Management Console 3.0 (MMC 3.0). MMC is a framework that hosts the graphical interface and programming platform to create, open, and save the snap-ins.

The SnapManager snap-in allows you to manage the SnapManager application from Microsoft Management Console.

The user interface consists of three panes.

Scope pane: This is the left pane in the SnapManager graphical user interface. It lists SnapManager for instances.

- If you select SnapManager in the Scope pane, all of the actions pertaining to it are displayed in the Actions pane.
- If you select a specific server in the Scope pane, all the actions pertaining to it are displayed in the Actions pane.
- If you select Backup in the Scope pane, all the actions pertaining to backup operations are displayed in the Actions pane.
- If you select Restore in the Scope pane, all the actions pertaining to the restore operations are displayed in the Actions pane.
- If you select Scheduled Jobs in the Scope pane, all the actions pertaining to the scheduled jobs are displayed in the Actions pane.
- If you select Reports in the Scope pane, all the actions pertaining to reports are displayed in the Actions pane.

Result pane: This is the center pane in the SnapManager graphical user interface. It displays details of the type of instance that you select in the Scope pane.

Actions pane: This is the right pane in the SnapManager graphical user interface. It displays all of the actions that you can perform, based on the instance that you select in the Scope pane.

Possible actions in the Actions pane include the following:

- Select the server and then click *Configuration Wizard* in the Actions pane.

  Use this wizard to configure SnapManager databases, transaction logs, and SnapInfo directories.
- Select the server and then click *Configuration Wizard Options Settings* in the Actions pane.

  Use it to enable or disable database to be migrated to the local disk.

- Select the server and then click *Backup Wizard* in the Actions pane.

  Use this wizard to backup SnapManager databases and transaction logs.
- Select the server and then click *Backup Settings* in the Actions pane.

  Use this option to specify backup settings for databases and transaction logs.
- Select the server and then click *Backup Verification Settings* in the Actions pane.

  Use this option to specify backup verification, mount point and DBCC settings for databases, and transaction logs.
- Select the server and then click *Clone Wizard* in the Actions pane.

  Use this option to clone existing backup sets and active production database. You can also delete clones and specify post restore settings.
- Select the server and then click *Run Command After Operation* in the Actions pane.

  Use this option to automatically run your own program or script after a backup or database verification operation finishes. It is typically used to automatically archive a backup.
- Select *Delete backup* in the Actions pane and select the backups that you want to delete.

  Use this feature to delete SnapManager backups that you do not want to restore, based on the number of backups, the retention period, or the type of backups.
- Select the server and then click *Restore Wizard* in the Actions pane.

  Use this wizard to restore from backups created on same server, on a different server, or from archived backup.
- Select *Fractional Space Reservation Settings* in the Actions pane to monitor the space reservation in the LUN and to set policy settings for LUNs and volumes.
- Select *Notification Settings* in the Actions pane to configure SMTP e-mail, storage system syslog, and AutoSupport for event notification.
- Select *Report Directory* Settings in the Actions pane to specify the report directory path for the server.
- Select *License Settings* in the Actions pane to update the SnapManager per-server license key.
- Select an server and then click *Disconnect Server* in the Actions pane to delete the Server from the SnapManager snap-in console.
- Select an Server and then click *Reconnect Server* in the Actions pane to reestablish the connection to the server.

- Select *Restore Setting* to configure recovery, restore, replication, and transaction log backup settings.

SnapManager Dashboard view

The Dashboard view enables you to view the status of different SnapManager for servers connected to the SnapManager for network. This is a dynamic view that gets refreshed frequently. Dashboard allows you to:

- View the server configuration
- Add new servers

SMSQL Server Configuration-Server Name: Click the server in the Scope pane to view Server Configuration details.

The following details are displayed in the Result pane:

- Name of the server instance
- Name of the host

Note

In the case of clustered configurations, this value must display the host name of the node to which SnapManager is connected.

- Server version
- SnapManager version
- Name of the verification server

Recent Operations: The following operations are listed:

- Last backup operation, including a time stamp and a hyperlink to the corresponding report
- Last restore operation, including a time stamp and a hyperlink to the corresponding report
- Last configuration operation, including a time stamp and a hyperlink to the corresponding report

Current Job Status: This view displays the jobs that are running and those in queue, for local as well as remote verification.

The following is the description of the elements of the pane:

- Status: It gives a link to the SnapManager report associated with the running job.
- Priority: It gives the queue position of the job.
- Job ID: It displays the Job ID of the job.

Note

During verification, SnapManager displays the parent Job ID of the job in the field Job ID. The parent Job ID is the ID the job had in the SQL server that sent the job to the verification server.

- State: It displays whether the job is running, cancelling, or in queue.
- Job Type: It displays whether the job is a backup job or backup with verification job. It can have any one of the following values:
  - Full Backup
  - Backup Verification
  - Restore
  - Test Restore
- Backup Source: It displays the name of the server that creates the backup.
- Submission time: It displays the time the job was submitted.
- Start time: It displays the time the backup job started.

Add new servers

You can add a new server from the Action pane and manage it through SnapManager. To add a new server, see “About the ‘Add Servers to be Managed’ option” on page 418.

Icons used in SnapManager

Icons used in SnapManager

The following table displays some icons frequently used by the SnapManager and a brief description of each.

Icon	Description
	◆ Add servers to be managed ◆ Reconnect server
	Backup settings
	Backup verification settings
	◆ Backup Wizard ◆ Backup and verify
	Configuration Wizard
	Configuration wizard option settings
	Delete backup
	Disconnect server
	Fractional space reservation settings
	License settings
	Notification settings

	Report directory settings
	Restore Wizard
	Run command after operation
	Databases selected for backup
	Unverified backup
	Verified backup

Configuration and volume mount points 6

About this chapter

This chapter describes SnapManager configuration and how volume mount point in SnapManager 5.0 helps you to surpass the 26-drive letter limitation. For SnapManager backup and restore to function, the data store must be located properly on one or more LUNs, and those LUNs need to be sized appropriately. The ability to restore individual databases imposes more data configuration requirements.

This chapter contains more information about SnapManager and SnapManager support for NTFS volume mountpoints. This information enables you to plan your SnapManager data store LUN and storage system volume requirements, based on your space and functionality requirements. The following topics are covered in this chapter:

The following topics are covered:

Preparing to Migrate SQL Server Databases

About this section

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

Related topics:

SQL Server configuration rules with SnapManager

About this section

This section outlines some of the rules governing configuration of SnapManager. Before you configure your SQL Server environment or migrate your database and transaction files to LUNs, ensure you have read and understood the following topics:

- "Maximum configurations supported by SnapManager" on page 95
- "SQL Server database configuration restrictions enforced" on page 96
- "SQL Server database configurations to avoid" on page 97

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 or Windows cluster	16
LUNs per SQL Server computer	120
Databases per LUN	35
Databases per storage system volume	35
Databases per SQL server instance	5000
File groups per database	75
Storage system volumes that can be used to store the following: ◆ A single database ◆ LUNs connected to an individual SQL Server computer	 30 120

Although SnapManager does not prevent you from creating configurations that exceed these limits, such configurations are untested and unsupported. It is best 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 98.

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 96
- "SQL Server database configurations to avoid" on page 97

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 99.
- An illustration of multiple databases sharing two LUNs is shown in "Multiple databases sharing two LUNs" on page 100.

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 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 the files* (the data files, in addition to 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, if those LUNs are not used for storing database files belonging to other databases. User database files and snapInfo directory cannot reside on the quorum disk.

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 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 LUN that hosts the SQL Server: You cannot migrate a user database to a LUN that hosts the SQL Server.

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 databases to LUNs, note the following recommendations against certain invalid configurations that could be created outside 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.

SQL Server configurations supported with SnapManager

About this section

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.

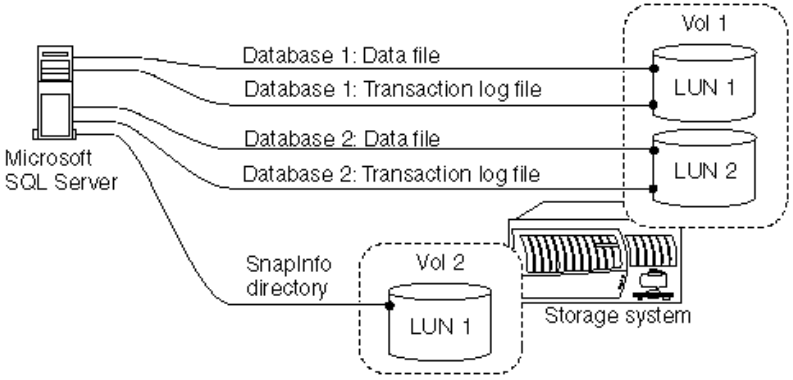
The supported configurations are as follows:

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, perform a backup immediately following any configuration changes.

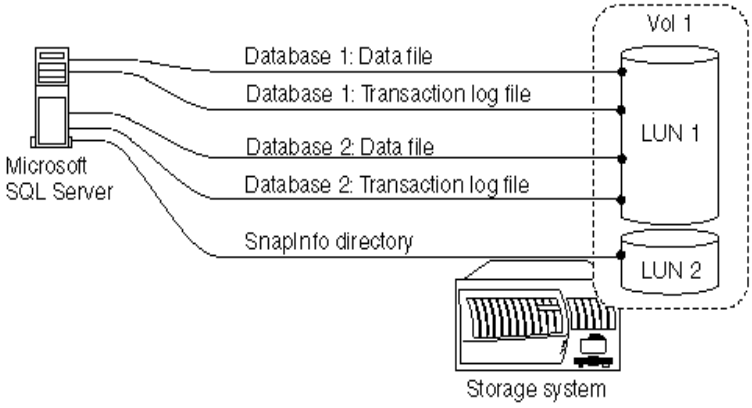
Multiple databases on different LUNs within the same volume

The following supported configuration shows multiple SQL Server databases sharing the same volume but residing on different LUNs.



Multiple databases on one LUN

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



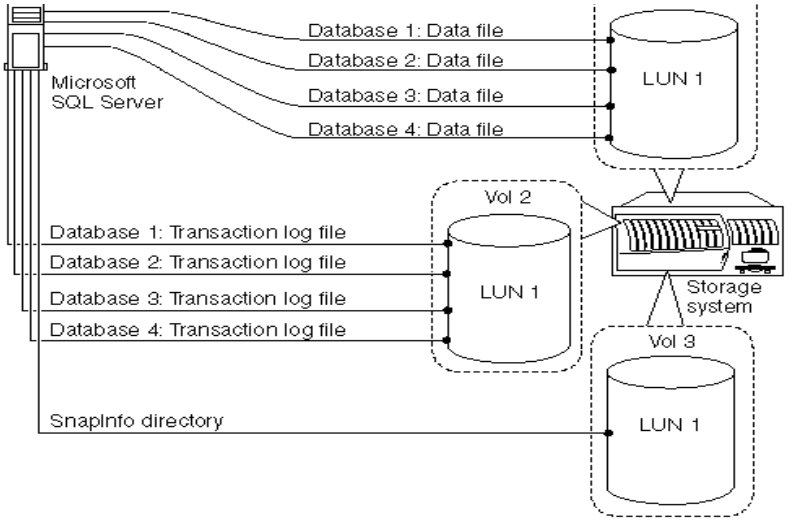
This is a simple configuration, and it can be applied to a SQL Server that supports about 35 databases per volume.

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. 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.



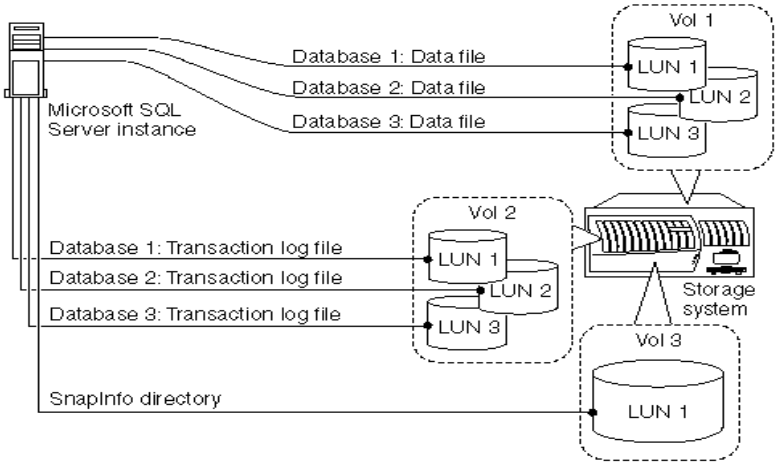
By placing the data files for multiple databases on one LUN and the transaction logs for those databases on the other LUN, SQL database performance is improved by separating the random I/O patterns of the data files from the sequential I/O patterns of the transaction log files.

Note

If you select 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 a 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.

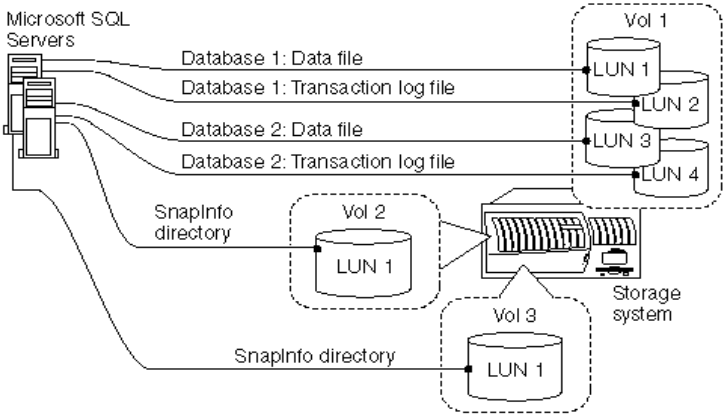


Multiple SQL Servers and one storage system volume

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:

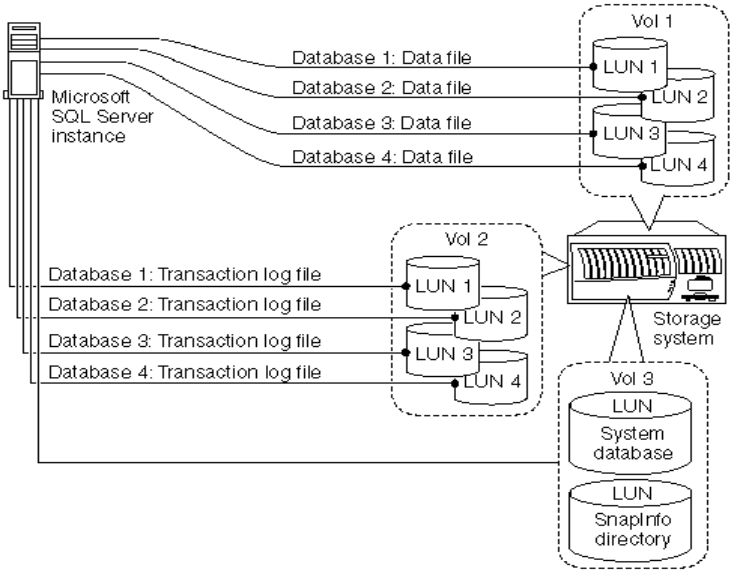
- If the volume fails, all databases are lost, including the current transaction log files.
- With a single volume housing databases for multiple SQL Server instances, there is an increased possibility of creating a busy Snapshot copy.

  For information about busy Snapshot copies, see "Busy Snapshot error prevents deletion of backup set" on page 222.



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.

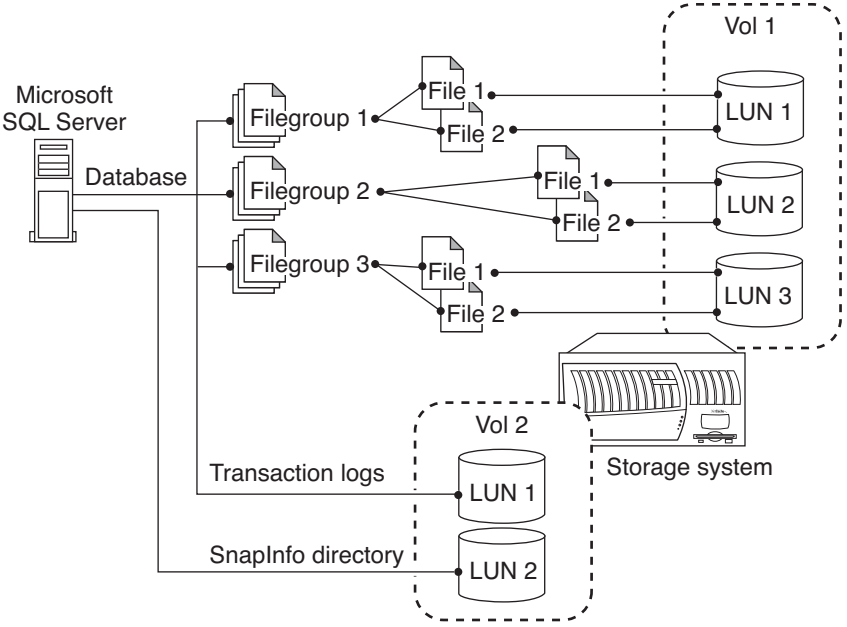


Note

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

Multiple file groups belonging to the same database on different LUNs

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



Understanding NTFS volume mount point

About this section

This section outlines the features and functionality of NTFS volume mount points:

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 you create a volume mount point on a server cluster, consider the following key items regarding volume mount points:
  - Volume mount points cannot be created between clustered and non clustered 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:

- 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 its disk source.

For more details, see the Microsoft TechNet article 280297.

Understanding SnapManager support for volume mount points

This topic describes SnapManager support for volume mount points:

Related topics:

- "Preparing to Migrate SQL Server Databases" on page 94

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. The restore operation is stream based, that is, only one database can be restored at a time. You can allocate a maximum of 25 drive letters.

Additionally, certain SnapManager operations require more drive letters for 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 5.0 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 in addition to those stored on standard Windows volumes.

Mounted volume environments supported by SnapManager

The following table summarizes the environments in which SnapManager 5.0 for SQL supports volumes mounted on LUNs. More details regarding limitations and enforcements imposed by SnapManager are described in subsequent sections of this document.

Windows host environment		Microsoft SQL Server		
		2000	**2005**	**2008**
2003	**Standalone**	Yes	Yes	Yes
	Clustered	No	Yes	Yes
2008	**Standalone**	--	Yes	Yes
	Clustered	--	Yes	Yes

Mounted volume restrictions with SnapManager

An NTFS volume that hosts mount points cannot support SQL Server databases. SnapManager imposes the following restrictions:

- SnapManager does not allow database files or database backup files to exist on an NTFS volume that has mount points.
- The mount point root LUN should not contain SQL database files, transaction log files or SnapInfo data files.

  The mount point root directory can also exist on your local disk.

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 an Available Disks selection are as follows:
  - Select a database, file or a file group to move to a LUN.
  - Setup a SnapInfo directory for all databases
  - Select a SnapInfo directory for each server instance

  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 Available Disks selection contains two entries for one LUN:

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

Swap LUNs using a reference mount point: If a database resides in LUN M, create a reference `C:\Mnt_Pnt1\db` to it using SnapDrive. You use the Configuration wizard to migrate the database from the original location LUN M 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.

  SnapManager configuration wizard lists all references to the same LUN. In this case, the Available Disks 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 Available Disks 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.

- To specify which method is to be used to access database backup sets during database integrity verification, use the Mount Point option 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 > Backup Verification Settings
  - Backup Wizard > Verification Settings
  - Restore Wizard > Verification Settings

  For more information, see Appendix C, “Using the Mount Point tab,” on page 425.

Backup and recovery using volume mount point

Backup and recovery using volume mount point

The following topics are covered:

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, see "Using the SnapManager Configuration Wizard" on page 121.
2. Create a backup of all the databases residing on volume mount point. For more information, see "Backing Up Databases Using SnapManager" on page 173.

Note

In SQL Server, the transaction log backups are stored in dump files which 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, see "Restoring Databases Using SnapManager" on page 225.

Change backup management group with mounted volume

To delete the backup set that resides on mounted volume, the following is an overview of the tasks you need to complete:

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 option and delete the databases. SnapManager should be able to delete Snapshot copies and backup metadata that resides on the mounted volume.

Developing your SnapManager data configuration plan

Developing your SnapManager data configuration plan

After you determine how many LUNs you need for your SnapManager configuration and what data those LUNs hold, use the information in this section to develop your SnapManager data configuration plan and prepare storage system volumes and LUNs for use with SnapManager. This entails calculating and recording the required sizes for the LUNs, which also determines the sizes of the volumes that contain the LUNs. Use the information you record in your SnapManager data configuration plan to create or modify the storage system volumes and LUNs on your storage system.

The information you record in your SnapManager data configuration plan could be used if problems arise later with your system. Knowing the volume and drive for each of your LUNs can aid the diagnosis and resolution of many potential issues.

To create your SnapManager data configuration plan, complete the following steps.

Step	Action
1	For each LUN you need, record the following information: ◆ Purpose ◆ Size ◆ Volume and qtree ◆ Assigned drive letter or mountpoint For details about calculating LUN size for a database, see "LUN size calculations" on page 115

Step	Action
2	For each volume you need to store LUNs, record the following information: ◆ Location (storage system name) ◆ Purpose ◆ Type (traditional or flexible) ◆ Fractional reserve (%) ◆ Automatic Snapshot copy deletion setting (enabled or disabled) ◆ LUNs contained ◆ Volume autogrow (enabled or disabled)

Assessing volume size

The following topics describe how to estimate the storage requirements on your storage system:

- "Storage system volume sizing requirements" on page 113
- "LUN size calculations" on page 115
- "Overall storage system volume requirements for a transaction log" on page 116
- "Criteria for estimating the amount of space required for a transaction log" on page 116
- "Initial sizing guidelines for new environments" on page 117

For more 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 more 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.

  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 days' worth of Snapshot copies are online at the same time.

  The first Snapshot copy lock uses disk space equal to the LUN object size itself (therefore, double the requirement). More 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.
- You only want to expand the volume every six months.
- Performance is important, that 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 about 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:

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

Criteria for 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 increases a transaction log file's size.

Overview of the database migration procedure

The following steps summarize the migration of SQL Server 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 backup 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 5.0 provides the capability to back up 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 the migration process, SnapManager for SQL skips this option only for the 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 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.

Prerequisites for migrating databases to LUNs

Before you migrate your SQL Server databases, you must 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 database
  - One or more LUNs for the SnapInfo files

  For resource planning information, see Chapter 6, "Preparing to Migrate SQL Server Databases," on page 94. For detailed instructions about 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 select that users are not connected to a database, see your SQL Server documentation.
- The SQL Server databases to be migrated and the 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 97.

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 the 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.

Using the SnapManager Configuration Wizard 7

About this chapter

SnapManager Configuration feature enables you to select the database verification servers, move the databases and transaction logs to LUNs, and configure automatic event notification.This chapter describes the following:

- How to plan your storage system layout
- How to configure your data using SnapManager

The following topics are covered in this chapter:

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 messages at the system console.

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 cannot be integrated to more than two LUNs cannot be used.

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. Transaction logs that belong to more than two LUNs cannot be used.

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, the Configuration wizard 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.
   - 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
4. After the migration is complete, the user databases are brought back online

Understanding the Configuration wizard

About this section

The primary function of the Configuration Wizard is to migrate SQL Server databases to 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 123
- "What the Configuration Wizard does not do" on page 124
- "When to use the Configuration Wizard" on page 124
- "Settings configurable only with the Configuration Wizard" on page 126
- "Settings configurable only with the Configuration Wizard" on page 126

What the Configuration Wizard does

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

From local disk to LUN: To enable management by SnapManager.

- If databases need to be moved, the wizard dismounts the databases, moves the database and transaction log files to the selected LUN, and remounts the databases.

  Note
  SnapManager takes databases offline during the move operation.

- The wizard creates a SnapInfo directory that SnapManager uses to store information about the backup sets and the backed-up transaction logs.
- The wizard disables circular logging for all databases that are moved to a LUN, to enable up-to-the-minute recovery of those databases.
- The wizard also guides you through several application settings. These settings include enabling notification of SnapManager events using e-mail, and enabling notification of SnapManager events using the storage system Syslog or the AutoSupport feature.

From LUN to LUN: If resource management issues require this. For example, consolidating SQL server on another storage system.

From LUN to local disk: Even if the databases are no longer managed using SnapManager, you can migrate them back to your local drive.

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.

Note

If you are migrating ReportServer database from your local disk to a LUN, ensure that SQL Server Agent and SQL Server Reporting Services are not running.

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.

Attention

Use the SnapManager Configuration wizard to move databases, transaction logs, or database system files. SnapManager ensures that these files are place in locations that meet SnapManager configuration requirements. Incorrectly located database, transaction logs, or database system files impair SnapManager operation. If some other medium is used to move the database, transaction logs, or database system files, run the SnapManager Configuration wizard after the operation to ensure that these files are in correct locations.

When to re-run the Configuration Wizard

Whenever you change the configuration manually using SnapDrive or the System Manager, you need to re-run the Configuration wizard to inform SnapManager that configuration has been modified. For example, when you manually change the SnapManager configuration from using volume mountpoints to drive letters, or from using drive letters to volume mountpoints, you need to run the Configuration wizard again for the changes to take effect. Another example of when the Configuration wizard needs to be run is when a new database is added using the SQL Management Studio.

About SnapManager components

When you use the Configuration wizard, you are specifying the placement of several components of SQL Server and SnapManager.

Server instance system path: Path to the directory that contains the server instance files. SnapManager places instance system files on the same LUN that stores the transaction logs for that database. The server instance system files must remain where the Configuration wizard places them.

Server instance databases: The databases in a database or store. The .mdf file is the primary database file. The .ndf file is the secondary database file.

Transaction log files: The location of the transaction logs. The transaction logs contain changes made to the databases since the last backup, enabling an up-to-the-minute restore. The .ldf file is the transaction log file.

SnapInfo directory: Contains SnapManager backup information, copies of transaction log files, and other data critical to the backup set.

Settings configurable only with the Configuration Wizard

The following table lists the 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	Select a database to move to a LUN
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 **Note** If an FCP or iSCSI hardware initiator is present on your system, then the option to add Microsoft iSCSI Service as a dependency is displayed as inactive.

Understanding control-file based configuration

About this section

This section describes how to use control-file based configuration to configure basic SnapManager settings. The following topics are covered in this section:

About control-file

The control-file is an XML file that contains SnapManager configuration information. The configuration data is represented in XML format. It can be edited manually.

Note

To avoid syntax errors, use an XML editor to edit the control-file configuration.

You can access the control-file option from the SnapManager Configuration wizard. You can use the control-file as an alternative to the SnapManager configuration wizard to configure SnapManager. This is especially useful in the following scenarios:

- Multiple SQL Server database servers, databases, and LUNs
- Disaster recovery
- Mass deployment

The configuration settings contained in the control-file are grouped into the following sections:

- Storage layout
- Notification settings
- Verification settings
- Report directory setting
- Backup settings
- SnapMirror relationship settings
- Scheduled jobs

Importing and exporting configuration settings

The following tasks can be performed using the control-file:

- Export the current configuration details to a control-file.
- Export a specific section of current configuration to a control-file.
- Import configuration details from a control-file.
- Import a specific section of configuration information from a control-file.

To import or export configuration settings, complete the following steps:

Step	Action
1	If you have not already done so, start SnapManager by accessing the Windows Start menu, and selecting Program Files > NetApp> SnapManager for SQL Server. **Result:** The SnapManager for SQL Server console appears.
2	In the Scope pane, double-click SnapManager for SQL Server. **Result:** SnapManager displays the Status dashboard in the Result pane.
3	Click the SQL Server database server that you want to configure.
4	In the Actions pane, click Configuration Wizard. **Result:** The Configuration Wizard launches and the Welcome window appears.
5	Select the “Use control-file” check box.
6	Click Next. **Result:** The Import or Export Selection window appears.

Step	Action
7	Select either the Import or the Export option. Selecting the Import option will enable the Review current settings in the Configuration wizard check box. Select this check box to review imported settings in the configuration wizard. If you have selected the Import option and unchecked the Review settings in the configuration wizard, SnapManager will proceed to the normal configuration wizards for you to confirm the imported configuration settings. Selecting the Export option causes the Review current settings in the Configuration wizard check box to be grayed out. If you selected the Export option, SnapManager exports the current configuration and settings to the control-file.
8	In the Use control-file check box, select the control-file path. SnapManager uses the default path `C:\Program Files\NetApp\SnapManager for SQL Server\SMSQLConfig_mm_dd_yyyy_hh.mm.ss.xml`.
9	Click Advanced.
10	In the Configuration Import/Export Advanced options window, specify the configuration settings that need to be imported or exported.
11	Click OK to confirm the configuration specification or Cancel to go back to the Import or Export Selection window. If SnapManager detects that there is some missing data in any of the selected options, it prompts you if you still want to carry out with the configuration.
12	Click Next to proceed. **Result:** The Verification Settings screen appears.
13	Select the verification server and the connection to be used. The connection can be Windows Authentication or SQL Server Authentication.

Step	Action
14	If you selected SQL Server Authentication, enter the login name and password. **Result:** SnapManager loads the control-file and validates the imported configuration and settings.

Sample XML schema for the control-file settings

The SnapManager schema file is distributed with the installation package. The following configuration file depicts the SnapManager control-file settings.

Storage layout settings: The following schema depicts the storage layout settings section. You can edit the storage layout settings using an XML editor.

```
<?xml version="1.0" ?>
- <SMSQLCONFIG xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<HOST_NAME>SNAPMGR-19</HOST_NAME>
- <STORAGE_LAYOUT>
<MAX_DB_JOB>255</MAX_DB_JOB>
- <SQL_INSTANCES>
- <SQL_INSTANCE>
<SQL_INSTANCE_NAME>SNAPMGR-19</SQL_INSTANCE_NAME>
<SQL_INSTANCE_SNAPINFO_PATH>K:\SMSQL_SnapInfo</SQL_INSTANCE_SNAPIN
FO_PATH>
<ADD_MSISIC_DEPENDENCY>false</ADD_MSISIC_DEPENDENCY>
- <DATABASES>
- <DATABASE>
<DATABASE_NAME>master</DATABASE_NAME>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>master</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\master.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>mastlog</FILE_NAME>
```

```
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\mastlog.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>tempdb</DATABASE_NAME>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>tempdev</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA
empdb.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>templog</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA
emplog.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>model</DATABASE_NAME>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>modeldev</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\model.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>modellog</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\modellog.ldf</FILE_PATH>
</LOG_FILE>
```

```
</LOG_FILES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>msdb</DATABASE_NAME>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>MSDBData</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\MSDBData.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>MSDBLog</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\MSDBLog.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>DB1</DATABASE_NAME>
<SNAPINFO_PATH>K:\SMSQL_SnapInfo</SNAPINFO_PATH>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>DB1</FILE_NAME>
<FILE_PATH>K:\MP\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB1.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>DB1_log</FILE_NAME>
<FILE_PATH>K:\MP2\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB1_log.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
- <DB_VOLUMES>
```

```
- <DB_VOLUME>
<FILER_NAME>rhine</FILER_NAME>
<VOLUME_NAME>grace2</VOLUME_NAME>
</DB_VOLUME>
</DB_VOLUMES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>DB3</DATABASE_NAME>
<SNAPINFO_PATH>K:\SMSQL_SnapInfo</SNAPINFO_PATH>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>DB3</FILE_NAME>
<FILE_PATH>I:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB3.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>DB3_log</FILE_NAME>
<FILE_PATH>I:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB3_log.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
- <DB_VOLUMES>
- <DB_VOLUME>
<FILER_NAME>rhine</FILER_NAME>
<VOLUME_NAME>grace1</VOLUME_NAME>
</DB_VOLUME>
</DB_VOLUMES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>DB2</DATABASE_NAME>
<SNAPINFO_PATH>K:\SMSQL_SnapInfo</SNAPINFO_PATH>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>DB2</FILE_NAME>
<FILE_PATH>K:\MP2\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB2.mdf</FILE_PATH>
</DATABASE_FILE>
```

```
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>DB2_log</FILE_NAME>
<FILE_PATH>K:\MP\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB2_log.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
- <DB_VOLUMES>
- <DB_VOLUME>
<FILER_NAME>rhine</FILER_NAME>
<VOLUME_NAME>grace2</VOLUME_NAME>
</DB_VOLUME>
</DB_VOLUMES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>DB4</DATABASE_NAME>
<SNAPINFO_PATH>K:\SMSQL_SnapInfo</SNAPINFO_PATH>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>DB4</FILE_NAME>
<FILE_PATH>I:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB4.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>DB4_log</FILE_NAME>
<FILE_PATH>I:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB4_log.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
- <DB_VOLUMES>
- <DB_VOLUME>
<FILER_NAME>rhine</FILER_NAME>
<VOLUME_NAME>grace1</VOLUME_NAME>
</DB_VOLUME>
</DB_VOLUMES>
</DATABASE>
- <DATABASE>
```

```
<DATABASE_NAME>DB5</DATABASE_NAME>
<SNAPINFO_PATH>K:\SMSQL_SnapInfo</SNAPINFO_PATH>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>DB5</FILE_NAME>
<FILE_PATH>I:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB5.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>DB5_log</FILE_NAME>
<FILE_PATH>I:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\DB5_log.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
- <DB_VOLUMES>
- <DB_VOLUME>
<FILER_NAME>rhine</FILER_NAME>
<VOLUME_NAME>grace1</VOLUME_NAME>
</DB_VOLUME>
</DB_VOLUMES>
</DATABASE>
</DATABASES>
</SQL_INSTANCE>
- <SQL_INSTANCE>
<SQL_INSTANCE_NAME>SNAPMGR-19\MARS</SQL_INSTANCE_NAME>
<SQL_INSTANCE_SNAPINFO_PATH>K:\SMSQL_SnapInfo</SQL_INSTANCE_SNAPIN
FO_PATH>
<ADD_MSISIC_DEPENDENCY>false</ADD_MSISIC_DEPENDENCY>
- <DATABASES>
- <DATABASE>
<DATABASE_NAME>master</DATABASE_NAME>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>master</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.2\MSSQL\DATA\master.mdf</FILE_PATH>
</DATABASE_FILE>
```

```
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>mastlog</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.2\MSSQL\DATA\mastlog.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>tempdb</DATABASE_NAME>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>tempdev</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL Server\MSSQL.2\MSSQL\DATA
empdb.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>templog</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL Server\MSSQL.2\MSSQL\DATA
emplog.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>model</DATABASE_NAME>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>modeldev</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.2\MSSQL\DATA\model.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
```

```
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>modellog</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.2\MSSQL\DATA\modellog.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
</DATABASE>
- <DATABASE>
<DATABASE_NAME>msdb</DATABASE_NAME>
- <FILE_GROUPS>
- <FILE_GROUP>
<GROUP_NAME>PRIMARY</GROUP_NAME>
- <DATABASE_FILES>
- <DATABASE_FILE>
<FILE_NAME>MSDBData</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.2\MSSQL\DATA\MSDBData.mdf</FILE_PATH>
</DATABASE_FILE>
</DATABASE_FILES>
</FILE_GROUP>
</FILE_GROUPS>
- <LOG_FILES>
- <LOG_FILE>
<FILE_NAME>MSDBLog</FILE_NAME>
<FILE_PATH>C:\Program Files\Microsoft SQL
Server\MSSQL.2\MSSQL\DATA\MSDBLog.ldf</FILE_PATH>
</LOG_FILE>
</LOG_FILES>
</DATABASE>
</DATABASES>
</SQL_INSTANCE>
</SQL_INSTANCES>
</STORAGE_LAYOUT>
```

Notification settings: The following schema depicts the notification settings section. You can edit the notification settings using an XML editor.

```
-<COMMON_SETTINGS>
-<NOTIFICATION>
-<SEND_EMAIL_NOTIFICATION>
<SMTP_SERVER>SNAPMGR-19</SMTP_SERVER>
<FROM>SMSQLAutoSender</FROM>
<TO>gracew@netapp.com</TO>
<SUBJECT>SnapManager for SQL Server</SUBJECT>
<NOTIFY_AUTO>true</NOTIFY_AUTO>
<LONG_MSG>false</LONG_MSG>
<AS_ATTACHMENT>false</AS_ATTACHMENT>
```

```
<SEND_ON_FAILURE>true</SEND_ON_FAILURE>
</SEND_EMAIL_NOTIFICATION>
<EMS_ENABLED>true</EMS_ENABLED>
<ASUP_ENABLED>true</ASUP_ENABLED>
<ASUP_ON_FAIL>true</ASUP_ON_FAIL>
</NOTIFICATION>
```

Verification settings: The following schema depicts the verification settings section. You can edit the verification settings using an XML editor.

```
-<VERIFICATION>
-<VERIFICATION_CLIENT_SETTING>
<VERIFICATION_SERVER>SNAPMGR-19</VERIFICATION_SERVER>
<VER_SERVER_NTAUTH>true</VER_SERVER_NTAUTH>
<VER_DBCC_NOINDEX>false</VER_DBCC_NOINDEX>
<VER_DBCC_ALL_ERROR_MSG>false</VER_DBCC_ALL_ERROR_MSG>
<VER_DBCC_NO_INFO_MSGS>false</VER_DBCC_NO_INFO_MSGS>
<VER_DBCC_TABLOCK>false</VER_DBCC_TABLOCK>
<VER_DBCC_PHYSICAL_ONLY>false</VER_DBCC_PHYSICAL_ONLY>
<VER_DBCC_ATTACH_DB>false</VER_DBCC_ATTACH_DB>
<VER_DBCC_BEFORE_MIGRATION>true</VER_DBCC_BEFORE_MIGRATION>
<VER_DBCC_AFTER_MIGRATION>false</VER_DBCC_AFTER_MIGRATION>
<VER_DELETE_DB_FILE_ORIG>true</VER_DELETE_DB_FILE_ORIG>
<VER_RUN_UPDATE_STATISTICS>true</VER_RUN_UPDATE_STATISTICS>
</VERIFICATION_CLIENT_SETTING>
-<VERIFICATION_SERVER_SETTING>
<AUTO_DRIVELETTER>true</AUTO_DRIVELETTER>
<MP_DIR>C:\Program Files\NetApp\SnapManager for SQL
Server\SnapMgrMountPoint</MP_DIR>
</VERIFICATION_SERVER_SETTING>
</VERIFICATION>
```

Report directory settings: The following schema depicts the report directory settings section. You can edit the report directory settings using an XML editor.

```
<REPORT_DRIECTORY>C:\Program Files\NetApp\SnapManager for SQL
Server\Report</REPORT_DRIECTORY>
```

Backup settings: The following schema depicts the backup settings section. You can edit the backup settings using an XML editor.

```
-<BACKUP>
-<BACKUP_CLIENT_SETTING>
<NAMING_CONVENTION>0</NAMING_CONVENTION>
<BACKUP_SET_TO_KEEP>3</BACKUP_SET_TO_KEEP>
<BACKUP_SET_TO_KEEP_IN_DAYS>0</BACKUP_SET_TO_KEEP_IN_DAYS>
<DELETE_BACKUPS_OPTION>0</DELETE_BACKUPS_OPTION>
<BACKUP_SET_TO_VERIFY>0</BACKUP_SET_TO_VERIFY>
</BACKUP_CLIENT_SETTING>
-<BACKUP_SERVER_SETTING>
```

```
<RUN_CMD_HOST>SNAPMGR-19</RUN_CMD_HOST>
<RUN_CMD_PATH>notepad.exe</RUN_CMD_PATH>
<RUN_CMD_ARGUMENT>$SqlSnapshot $InfoSnapshot</RUN_CMD_ARGUMENT>
</BACKUP_SERVER_SETTING>
</BACKUP>
```

SnapMirrored volumes settings: The following schema depicts the SnapMirror relationship settings section. You can edit the SnapMirror relationship settings using an XML editor.

```
-<VERIFICATION_ON_DESTINATION>
-<SELECTED_DESTINATIONS>
-<SELECTED_DESTINATION>
<SOURCE_FILER>rhine</SOURCE_FILER>
<SOURCE_VOLUME>grace2</SOURCE_VOLUME>
<DESTINATION_FILER>rhine</DESTINATION_FILER>
<DESTINATION_VOLUME>grace2_mir</DESTINATION_VOLUME>
</SELECTED_DESTINATION>
-<SELECTED_DESTINATION>
<SOURCE_FILER>rhine</SOURCE_FILER>
<SOURCE_VOLUME>grace2</SOURCE_VOLUME>
<DESTINATION_FILER>rhine</DESTINATION_FILER>
<DESTINATION_VOLUME>grace2_mir</DESTINATION_VOLUME>
</SELECTED_DESTINATION>
</SELECTED_DESTINATIONS>
</VERIFICATION_ON_DESTINATION>
```

Schedule job settings: The following schema depicts the schedule job settings section. You can edit the schedule job settings using an XML editor.

```
-<VERIFICATION_ON_DESTINATION>
-<SELECTED_DESTINATIONS>
-<SELECTED_DESTINATION>
<SOURCE_FILER>rhine</SOURCE_FILER>
<SOURCE_VOLUME>grace2</SOURCE_VOLUME>
<DESTINATION_FILER>rhine</DESTINATION_FILER>
<DESTINATION_VOLUME>grace2_mir</DESTINATION_VOLUME>
</SELECTED_DESTINATION>
-<SELECTED_DESTINATION>
<SOURCE_FILER>rhine</SOURCE_FILER>
<SOURCE_VOLUME>grace2</SOURCE_VOLUME>
<DESTINATION_FILER>rhine</DESTINATION_FILER>
<DESTINATION_VOLUME>grace2_mir</DESTINATION_VOLUME>
</SELECTED_DESTINATION>
</SELECTED_DESTINATIONS>
</VERIFICATION_ON_DESTINATION>
-<SCHEDULE_JOBS>
-<JOB>
<SCHEDULER>Windows Task Scheduler</SCHEDULER>
```

```
<JOB_NAME>bkup1</JOB_NAME>
<HOST_NAME>snapmgr-19</HOST_NAME>
<START_DIR>C:\Program Files\NetApp\SnapManager for SQL
Server\</START_DIR>
<APPLICATION_NAME>C:\Program Files\NetApp\SnapManager for SQL
Server\SMSQLJobLauncher.exe</APPLICATION_NAME>
<COMMAND>new-backup ñsvr 'SNAPMGR-19' -db 'SNAPMGR-19', '8', 'DB1',
'DB2', 'DB3', 'DB4', 'DB5', 'master', 'model', 'msdb', 'SNAPMGR-
19\MARS', '3', 'master', 'model', 'msdb' -ver ñversvr 'SNAPMGR-19'
-del -rtbkups 2 -lgbkafbk -noutm -uniq ñmgmt standard</COMMAND>
<START_TIME>11/6/2007 1:32:00 PM</START_TIME>
-<SCHEDULES>
-<WEEKLY_TRIGGERS>
-<WEEKLY_TRIGGER>
-<TASK_TRIGGER>
<TriggerSize>48</TriggerSize>
<Reserved1>0</Reserved1>
<BeginYear>2007</BeginYear>
<BeginMonth>10</BeginMonth>
<BeginDay>27</BeginDay>
<EndYear>0</EndYear>
<EndMonth>0</EndMonth>
<EndDay>0</EndDay>
<StartHour>13</StartHour>
<StartMinute>32</StartMinute>
<MinutesDuration>0</MinutesDuration>
<MinutesInterval>0</MinutesInterval>
<Flags>0</Flags>
<Type>TIME_TRIGGER_WEEKLY</Type>
-<Data>
-<daily>
<DaysInterval>1</DaysInterval>
</daily>
-<weekly>
<WeeksInterval>1</WeeksInterval>
<DaysOfTheWeek>4</DaysOfTheWeek>
</weekly>
-<monthlyDate>
<Days>262145</Days>
<Months>0</Months>
</monthlyDate>
-<monthlyDOW>
<WhichWeek>1</WhichWeek>
<DaysOfTheWeek>4</DaysOfTheWeek>
<Months>0</Months>
</monthlyDOW>
</Data>
```

```
<Reserved2>0</Reserved2>
<RandomMinutesInterval>0</RandomMinutesInterval>
</TASK_TRIGGER>
</WEEKLY_TRIGGER>
</WEEKLY_TRIGGERS>
</SCHEDULES>
</JOB>
-<JOB>
<SCHEDULER>SQL Server Agent</SCHEDULER>
<JOB_NAME>bkupSqlAgt</JOB_NAME>
<HOST_NAME>SNAPMGR-19</HOST_NAME>
<START_DIR>C:\Program Files\NetApp\SnapManager for SQL
Server\</START_DIR>
<APPLICATION_NAME>C:\Program Files\NetApp\SnapManager for SQL
Server\SMSQLJobLauncher.exe</APPLICATION_NAME>
<COMMAND>ackup ñsvr 'SNAPMGR-19' -db 'SNAPMGR-19', '8', 'DB1',
'DB2', 'DB3', 'DB4', 'DB5', 'master', 'model', 'msdb', 'SNAPMGR-
19\MARS', '3', 'master', 'model', 'msdb' -ver ñversvr 'SNAPMGR-19'
-del -rtbkups 2 -lgbkafbk -noutm -uniq ñmgmt standard</COMMAND>
<START_TIME>11/7/2007 1:00:00 AM</START_TIME>
-<SQLAGENTSCHEDULES>
<START_DATE_TIME>11/5/2007 12:00:00 AM</START_DATE_TIME>
<START_TIME_OF_DAY>01:00:00</START_TIME_OF_DAY>
<END_DATE_TIME>12/31/9999 12:00:00 AM</END_DATE_TIME>
<END_TIME_OF_DAY>23:59:59</END_TIME_OF_DAY>
<FREQUENCY_TYPE>Daily</FREQUENCY_TYPE>
<FREQUENCY_INTERVAL>1</FREQUENCY_INTERVAL>
<FREQUENCY_SUBDAY_TYPE>Once</FREQUENCY_SUBDAY_TYPE>
<FREQUENCY_SUBDAY_INTERVAL>0</FREQUENCY_SUBDAY_INTERVAL>
<FREQUENCY_RELATIVE_INTERVAL>First</FREQUENCY_RELATIVE_INTERVAL>
<FREQUENCY_RECURRENCE_FACTOR>0</FREQUENCY_RECURRENCE_FACTOR>
</SQLAGENTSCHEDULES>
</JOB>
-<JOB>
<SCHEDULER>SQL Server Agent</SCHEDULER>
<JOB_NAME>bkupSqlAgtMars</JOB_NAME>
<HOST_NAME>SNAPMGR-19\MARS</HOST_NAME>
<START_DIR>C:\Program Files\NetApp\SnapManager for SQL
Server\</START_DIR>
<APPLICATION_NAME>C:\Program Files\NetApp\SnapManager for SQL
Server\SMSQLJobLauncher.exe</APPLICATION_NAME>
<COMMAND>ackup ñsvr 'SNAPMGR-19' -db 'SNAPMGR-19', '8', 'DB1',
'DB2', 'DB3', 'DB4', 'DB5', 'master', 'model', 'msdb', 'SNAPMGR-
19\MARS', '3', 'master', 'model', 'msdb' -ver ñversvr 'SNAPMGR-19'
-del -rtbkups 2 -lgbkafbk -noutm -uniq ñmgmt standard</COMMAND>
<START_TIME>11/11/2007 2:00:00 AM</START_TIME>
-<SQLAGENTSCHEDULES>
```

```
<START_DATE_TIME>11/5/2007 12:00:00 AM</START_DATE_TIME>
<START_TIME_OF_DAY>02:00:00</START_TIME_OF_DAY>
<END_DATE_TIME>12/31/9999 12:00:00 AM</END_DATE_TIME>
<END_TIME_OF_DAY>23:59:59</END_TIME_OF_DAY>
<FREQUENCY_TYPE>Weekly</FREQUENCY_TYPE>
<FREQUENCY_INTERVAL>1</FREQUENCY_INTERVAL>
<FREQUENCY_SUBDAY_TYPE>Once</FREQUENCY_SUBDAY_TYPE>
<FREQUENCY_SUBDAY_INTERVAL>0</FREQUENCY_SUBDAY_INTERVAL>
<FREQUENCY_RELATIVE_INTERVAL>First</FREQUENCY_RELATIVE_INTERVAL>
<FREQUENCY_RECURRENCE_FACTOR>1</FREQUENCY_RECURRENCE_FACTOR>
</SQLAGENTSCHEDULES>
</JOB>
</SCHEDULE_JOBS>
</COMMON_SETTINGS>
</SMSQLCONFIG>
```

Migrating SQL Server databases to LUNs

Migrating databases to LUNs using the Configuration Wizard

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

Attention

After a 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 application. To do this, go to the Windows Start menu and select Program Files > NetApp > SnapManager for SQL Server. **Result:** The SnapManager for SQL Server console root opens.
2	Add a server. ◆ In the Actions pane, click 'Add Servers to be Managed'. Select a Server from the list, or use the Browse option to select a server, then click 'Add'. **Note** You can also add server instances using this option, as there can be many server instances within a domain or a single physical server. For more information, see "How to start SnapManager for the first time after installation" on page 79.
3	In the left pane, double-click the SnapManager hostname that you want to connect to.

Step	Action
4	The Configuration Wizard is launched and the Welcome screen appears. Click Next. **Result:** If you cleared the "Use control-file" option, the Database Verification Server and Account Authentication screen appears.
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 a SQL Server that is different from the verification server. ◆ If you want to specify the database verification server now, select a 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. ◆ 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 421.
6	Click Next. **Result:** The 'Database Selection' screen appears.
7	Follow the remaining screens as per the wizard.
8	Click Finish to complete the process.

Moving multiple SnapInfo directories to a single SnapInfo directory

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 (non-default) 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: ◆ Verification Settings and Account Authentication ◆ Database Selection
2	In the SnapInfo Settings 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 the current SnapInfo directories are selected by default.

Step	Action
3	In the Available Disks, select the LUN to which you want to move all the SnapInfo directories.
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`.
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.
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 Actions pane, select 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 "Select a database to move to a lun" screen of the Configuration Wizard does not list any local drives unless you explicitly enable the migration to a local disk feature.
2	Click OK.
3	Launch the SnapManager Configuration Wizard.
4	Click Next at the Welcome screen.
5	In the "Select a database to move to a lun" 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 re configured. **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 Select 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 more information about SnapManager Backup and SnapManager backup sets. The following topics are covered:

Note

If you are running SQL Server 2000, SnapManager does not allow you to restore a backup set created using SQL Server 2005 or SQL Server 2008.

How SnapManager Backup works

SnapManager Backup uses Snapshot copy functionality to create online, read-only copies of databases. After the selected databases are backed up, the transaction logs that are already committed to the databases captured in the backup are deleted.

Note

Databases that cannot be backed up by SnapManager are greyed out in the Results pane.

See the following topics for more information:

- "Types of backups SnapManager can perform" on page 150
- "What SnapManager Backup does" on page 150
- "SnapManager Backup requirements and limitations" on page 151

Types of backups SnapManager can perform

SnapManager Backup performs backups at the volume level:

Volume-wide backup: When a Snapshot copy is taken 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 that server. If data from other servers resides on the same volume, it is not restorable from that Snapshot copy.

Multiple-volume backups: SnapManager performs backups in parallel on all LUNs that belong to the same server and share a single storage system volume. When a database set spans multiple volumes, the resulting backup set contains multiple database Snapshot copies but is still restorable as a single entity.

Partial backups: If the backup of some of the databases in the database set fails, the databases in the set that were backed up successfully can still be restored. Because each database constitutes its own backup, or file, it is restored discretely, independent of backups of the others in its database set, even though the backup of all databases in the set was performed by the backup job.

What SnapManager Backup does

SnapManager 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 backup set of the LUNs containing the database files
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 backup set (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 use VFilers, ensure to enable the option `vfiler.vol_clone_zapi_allow` in SnapDrive which is disabled by default. If you do not enable this option, SnapDrive does not create clones for the database. For more information, see SnapDrive documentation.
- If you change the database configuration, any backups taken before the configuration change are invalid. After you run the Configuration wizard, immediately take a backup to reflect the change in your configuration.

Attention

In a Microsoft SQL Server 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.

How SnapManager backup data is organized

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

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:

- Databases
- Transaction logs
- SnapInfo directory

Note

SnapManager allows you to create backups for read-only databases also.

SnapInfo directory

The SnapInfo directory stores information about the streaming-based backups of system databases, copies of transaction log files, and the backup set's metadata. 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, in addition to 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.

Note

By default, the SnapInfo directory is on the LUN that stores the transaction log files, but this is not a requirement. The SnapInfo directory cannot reside on the same LUN that stores the database files.

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 that belong to the SQL Server instance `INST2` on the on the Windows host system `ENGR-WINSRVR7` would be named as follows: `SQL__INST2`

SnapManager backup set names

SnapManager backup set names identify the configuration of the backed-up databases. These names are displayed in the SnapManager Results pane 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 that belong 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 that belong to the SQL Server instance `INST2` on the on the Windows host system `ENGR-WINSRVR7` would be named as follows: `INST2`

SnapManager backup set naming conventions

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

SQL Server name: Database backup set 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 backup set 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 213.

Most recent backup: Earlier versions of SnapManager 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 identify and operate on the most recent backup set.

With the addition of the Run Command After Operation feature in SnapManager, appending `recent` is no longer necessary because the scripts can be integrated into the backup process.

SnapManager 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. This removes the need for the rename of the Snapshot copy when the next backup is created. This is the default naming convention for SnapManager 5.0.
- Generic backup naming

  The most recent Snapshot copy name contains the string `recent` instead of a date and time stamp. 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 older versions of SnapManager and is the default setting.

When you have dataset configured in your system, you can either choose to apply the unique backup naming convention with the archival process enabled, or to keep the generic backup naming convention. If you choose to keep the naming convention as generic, no archives of the database to be backed up at the remote location are created.

If you archive the backups using PowerShell, the generic backup naming convention is automatically changed to the unique backup naming convention.

The backup naming convention is selected in the Backup Settings dialog box. For information about using this dialog box, see "Configuring the profile of a full database backup" on page 427.

Note

You are advised to select the unique naming convention option unless you have legacy scripts that require the presence of a backup with "recent" in its name. You need to select the unique naming convention explicitly (using the Options > Backup Setting menu or the Backup Naming Convention screen of the Backup wizard) because for backward compatibility purposes, the generic naming convention is selected by default.

SQL Server database backup set names

For SQL Server, backup set names begin with the string `sqlsnap__`.

Backup management group	Format of the SQL Server database backup set 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`

Types of backup operations performed using SnapManager

SnapManager supports two types of backup operations:

- "Full database backup" on page 159
- "Transaction log backup" on page 161

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 that you select. The method that SnapManager uses to create the backup depends on the databases that 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 230.

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 backup 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.

If there is no system database on the LUN that hosts the SQL Server, SnapManager cannot backup the user databases on that LUN and displays those user databases as inactive.

Note

If there is a system database on the LUN that hosts the SQL Server, a user database can reside 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.

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 backup user databases that do not reside on the same LUN as 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 clear databases that reside on a different LUN from the databases you selected, even if the LUN is on the same storage volume. If the other LUN stores only a single database, you can clear or reselect that database individually. If the other LUN houses multiple databases, you must clear 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 155.

More about volume-wide backups: In a volume-wide backup, all the databases that reside on a single volume are backed up *concurrently* using Snapshot copies. Since the maximum number of databases supported per storage system volume is 35,

Total number of Snapshot copies created= Number of databases / 35

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 committed database changes that have occurred since the last transaction log backup that was truncated after the backup completed. SnapManager supports transaction log backups 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 "Ways to manage the number of backup sets kept online" on page 168.

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

- SnapManager full database backups include the option to also backup the associated transaction logs after the database Snapshot copy backups finish. This is described in "Creating a full database backup using SnapManager" on page 176.
- SnapManager also provides the option to backup transaction logs only, independent of the associated databases. This is described in "Creating a transaction log backup using SnapManager" on page 190.

About log shipping and other backup solutions: use SnapManager only, to backup your SQL Server database transaction log files. Snap Manager does support log shipping; therefore, 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 you use log shipping, you cannot backup the transaction logs for that database.

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

- When using SnapManager Backup, do not backup 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

SnapManager uses the `DBCC CHECKDB` command to verify the integrity of live databases in addition to 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.
- 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 427.

Verifying the integrity of databases in backup sets: Databases in backup sets can be verified on 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 Results pane lists the backups that have been taken and the backup verification status of each.

LUN 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 the database 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 is mounted requires one available drive letter or a mount point: To run the DBCC CHECKDB command, the verification server (whether local or remote) must have a sufficient number of drive letters available or a mount point 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* or a mount point 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 those 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 eight separate LUNs. In this case, the verification server would need to have a minimum of eight drive letters or a mount point 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 fails 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 176.
- Verification of the databases in the most recent unverified backup sets. For detailed information, see "Performing database verification using SnapManager" on page 202.

- Verification of the databases in an unverified backup set selected for a restore operation. For detailed information, see "Performing a restore operation" on page 237.

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 complete. 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 complete, you can disable that feature. You can then run a separate database verification operation any time after the full database backup operation is complete.

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 on 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 176.

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 202.

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 231.

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 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 37. The procedure specifying a different SQL Server as the remote verification server is described in "Selecting the database verification server" on page 421.

Note

You cannot run database verification from a virtual SQL Server. For more information, see "Requirements for a remote verification server" on page 37.

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*.

- To verify a SQL Server 2000 database, if you use another version SQL Server, 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 more processing overhead is incurred *each time* a SQL Server 2000 database backup set is verified using another version of SQL Server.

- If you use a *SQL Server 2000 verification server* to verify another version of SQL Server database, SQL Server DBCC will fail.
- If you select a *SQL Server 2005 or SQL Server 2008 database* for verification, SnapManager prevents you from selecting a SQL Server 2000 verification server for that operation.

Ways to manage the number of backup sets kept online

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

- "Maximum number of databases per LUN" on page 168
- "Automatic deletion of the oldest backups in a management group" on page 168
- "Explicit deletion of backup sets" on page 170

Maximum number of databases per LUN

SnapManager 5.0 supports a maximum of 35 databases per LUN.

Note

It is possible for the total number of backup sets 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.

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 backup sets of that database to be retained for that backup management group. After the backup is complete, SnapManager will automatically delete the oldest backup sets for that database in the specified backup management group, retaining only the number of backups you want to preserve, or the backups older than a specified number of days.

This is the recommended method for managing the number of backup sets you store on your system.

The procedural details are included in "Creating a full database backup using SnapManager" on page 176 and "Creating a transaction log backup using SnapManager" on page 190.

For more information about backup management groups, see “SnapManager backup set naming conventions” on page 155 and “Using backup management groups in backup and verification” on page 213.

Cases in which more 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 does 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

You do not need to perform an up-to-the-minute restore from the older backups, allow the logs to be deleted to free up more space on the storage system holding the backups.

Explicit deletion of backup sets

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

Explicit deletion of an individual backup: With this method, you delete individual selected backup sets for either full database backups or transaction logs. The procedures are described in "Busy Snapshot error prevents deletion of backup set" on page 222 and "Busy Snapshot error prevents deletion of backup set" on page 222.

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 backups" on page 218.

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 230. For a description of how to view and delete these Snapshot copies, see "Deleting restored Snapshot copies" on page 248.

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 171
- "Best time to run a SnapManager backup" on page 171
- "Frequency of backups" on page 171
- "Recommendations for scheduling backups" on page 172

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 222.
- 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 backup your SQL Server databases to SnapManager backup sets and how to verify databases stored in those backup sets. The following topics are covered:

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:

- "Understanding SnapManager Backup Sets" on page 149
- "Tools for Managing Backup and Verification" on page 359

How SnapManager backup functions are accessed

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

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

- "Backup and Verify" on page 174
- "SnapManager Backup wizard" on page 174
- "Default backup settings" on page 175
- "Default verification settings" on page 175

Backup and Verify

The SnapManager console root includes a Backup and Verify option that you can use to specify the job-specific parameters of a SnapManager backup operation or database verification.

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

- "Full database backup using Backup and Verify" on page 181
- "Transaction log backup using the Backup and Verify option" on page 193
- "Database verification using the Backup and Verify option" on page 205

SnapManager Backup wizard

An alternative to the Backup and Verify option, 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 185
- "Transaction log backup using the Backup wizard" on page 196
- "Database verification using the Backup wizard" on page 208

Default backup settings

The SnapManager for SQL Server-Backup 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 176
- "Information you need to specify for a transaction log backup" on page 190
- "Information you need to specify for a database verification" on page 202

For more information, see "SnapManager backup options" on page 427.

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 176
- "Information you need to specify for a database verification" on page 202

For more information, see "Database integrity verification options" on page 421.

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 Verify option. See the following topics for more information:

- "Information you need to specify for a full database backup" on page 176
- "Full database backup using Backup and Verify" on page 181
- "Full database backup using the Backup wizard" on page 185
- "Creating a transaction log backup using SnapManager" on page 190

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 option or in the Backup wizard.

- Which databases you want to backup

  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 clear automatically selected databases as follows:

  - You can clear databases that reside on a different LUN from the databases you selected, even if the LUN is on the same storage volume. If the other LUN stores only a single database, you can clear or re-select that database individually. If the other LUN houses multiple databases, you must clear or re-select those databases as a group.
  - In the case of a *stream-based* full database backup, you can clear any automatically selected database. 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 159.

  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 progresses as follows:

- The *backup report* includes 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 is logged to the *event log*. The description field of this event contains 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.
- When you click a database, SnapManager displays the following:

  The name of the database, the name of the LUN storage system, the name of the LUN volume that contains it, the SnapInfo directory and whether the database created using streaming or Snapshot technology.
- When backup is scheduled for future, existing backups are not deleted unless some of the databases in the server instance are first moved to the local disk.

  If the databases on the LUNs are moved to the local disk later, the scheduled backup operation skips backup deletion. The backup is re-created and rescheduled so that for future operations, backups are deleted first.

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.

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 213.
- The operation asks whether you want to automatically run a transaction log backup after the full database backup finishes.
- The operation asks whether you want to automatically delete the oldest full database backups within this backup management group (recommended 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 168.
- If you select to automatically delete the oldest full database backups within this backup management group: the operation asks 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 169.

- The operation asks whether you want to perform a database integrity verification of the backup set after the full database backup is complete

Note

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

- The operation asks whether you want to run a command after the backup is complete (usually done to archive backups)
- The operation asks 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 complete.

  For more information about this option, see "Replicating Backups to Mirrored Volumes" on page 267.
- The operation asks 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:

- The operation asks whether you want the backup set to be named using generic ("`__recent`") or unique (timestamped) naming conventions

  For more information, see "SnapManager backup set naming conventions" on page 155.
- The operation asks 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 option of the SnapManager for SQL Server-Backup dialog box, described in "Configuring the profile of a full database backup" on page 427.

The SnapManager for SQL Server-Backup dialog box can be accessed from the Backup and Verify option, as well as from the Backup wizard.

- If you are using the Backup and Verify option, you can open the SnapManager for SQL Server-Backup dialog box from the Actions pane.
- 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:

- The operation asks which SQL Server is used to perform database verification

  This is configured using the SQL Server option of the Verification Settings dialog box, described in "Selecting the database verification server" on page 421. 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 37.

  Note

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

  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 backup sets at the volume level or the SnapDrive level.

  For information about busy Snapshot copies, see "Busy Snapshot error prevents deletion of backup set" on page 222.
- The operation asks which DBCC options are used to verify database backup sets

  This is configured using the DBCC Options option of the Verification Settings dialog box, described in "Selecting DBCC options" on page 423.

The Verification Settings dialog box can be accessed from the Backup and Verify option, as well as from the Backup wizard:

- If you are using the Backup and Verify option, select “Verify most recent unverified Snapshot backups only” to open the Verification Settings tab.
- At least two of the most recent SnapManager Snapshot copies that were used for SnapVault updates need to be kept online on the primary storage system.

  You can use SnapManager to delete backup sets outside the backup process. When SnapVault is used, keep at least two of the most recent Snapshot copies used for the SnapVault updates until SnapVault updates for a given backup are complete. To check which Snapshot copies are required for the SnapVault relationships, use the Snapvault destinations -s command on the SnapVault primary storage system.

  When you use SnapManager to automatically delete older backup sets as part of a backup process, be sure to configure the "Delete backups In excess of" option to a number that is equal to or greater than two. If backups are automatically deleted based on time rather than quantity using the "Delete backups Older than" option, be sure to specify a number of days that will allow at least two of the Snapshot copies used for the SnapVault updates to remain online. This information also applies to deleting backups using the Delete Backup option from the SnapManager Action pane.

Note

This applies only to Snapshot copies that are used for SnapVault updates.

- From within the Backup wizard, you can open the Verification Settings dialog box by clicking “Verification Settings” button in the Verification Settings screen.

  Note

  The Verification Settings screen appears only if you are specifying a “Full database backup” or a “Verify most recent backup set 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 Backup and Verify

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

With a full database backup you can choose to also backup the associated transaction logs after the database backup is complete. If you want to backup only transaction logs, see "Creating a transaction log backup using SnapManager" on page 190.

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 176.

Step	Action
1	In the SnapManager console root, double-click the server you want to use.
2	Select the Backup option in the Scope pane.
3	In the Results pane, 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 storage system volume. For information about clearing any automatically selected databases, see the bullet "Which databases you want to backup" under "Job-specific parameters" on page 176. When you select databases at the *SQL Server instance level*, SnapManager reports 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 176.
4	In the Actions pane, select "Backup and Verify." **Result:** The "SnapManager for SQL Server-Backup" window appears.
5	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 backup set naming conventions" on page 155 and "Using backup management groups in backup and verification" on page 213.

Step	Action	
6	You can retain up-to-the-minute restore ability of older backups for other management groups by selecting “Retain up-to-the-minute restore ability of older backups in other management groups."	
7	**If...**	**Then...**
	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.
	You want to schedule the transaction log backup yourself later	See “Creating a transaction log backup using SnapManager” on page 190. **Note** When you schedule a transaction log backup, ensure that the full backup and transaction log backup do not coincide.
8	**If...**	**Then...**
	You want to delete backups older than a specific number of days	Enter the number of days in the “Older than” field
	You want to delete backups more than a specified number of backups	Enter the number of backups in the “In excess of” field

Step	Action
9	If you want to run a command or script after the backup finishes, select the “Run Command After Operation” option. This is done to archive backups. **Note** If you select this option, SnapManager prompts you for the details when you are ready to start or schedule the backup operation. For more information, see “Running a script from a UNC path on a Windows Server 2003 system” on page 360.
10	Under SnapMirror options, select the corresponding check box if you want to update SnapMirror after operation or to verify available SnapMirror destination volumes. **Note** If the system is configured for SnapMirror replication, only the databases housed on mirrored LUNs will be updated on the SnapMirror destination volumes.
11	If you want to verify databases after the backup operation, select “Verify databases after backup."
Back up now or schedule for later	
12	You can either run the backup now or schedule it for later. Click one of the following tabs: ◆ Backup Now ◆ Schedule
To complete this procedure by scheduling the backup:	
13	If you select “Schedule...”, complete the procedure by scheduling the backup. For details, see “Scheduling a backup job or a database verification job” on page 362.

Step	Action
To complete this procedure by starting the backup:	
14	If you select “Backup Now...” do the following: **a.** Read the items displayed in the Backup Task List. This list shows 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 complete. ◆ You can toggle the Backup Status dialog box between two different views—Backup Task List view and Backup Report 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, an e-mail is sent and the event is posted to the Windows Application event log.

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 176.

Step	Action
1	In the SnapManager for SQL Server console root, 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 Backup 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 storage system volume are shown with disk icons of the same color.
Databases to Backup or Verify	
3	In the left panel, click to select the databases you want to backup. 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 backup" under "Job-specific parameters" on page 176. When you select databases at the *SQL Server instance level*, SnapManager lists 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 176.
4	Click Next. **Result:** The Backup or Verify Databases and Transaction Logs screen appears.

Step	Action
Backup or Verify Databases and Transaction Logs	
5	Select the Backup Databases and Transaction Logs option, and then click Next. **Result:** The Backup Type screen appears. **Note** If you do not select the Backup Databases and Transaction Logs option and select to schedule the transaction log backup yourself, ensure that the full backup and transaction log backup operations do not coincide.
Select SQL Server backup Type	
6	Select the Full Database Backup option, and then click Next. **Result:** The "Select backup management group for this backup" screen appears.

Step	Action
Select backup management group for this backup	
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 ◆ Backup 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 when 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 176.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Backup Finish</td></tr>
<tr><td>8</td><td>The “Backup Finish” screen prompts you to choose whether you want the operation to be performed immediately 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 11.</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, Click Finish.

Result: The Backup Status screen appears and displays the backup settings you have selected.</td></tr>
<tr><td>10</td><td>In the Backup Status window, click Start Now.

Result: The backup is performed and the backup set is written to the volume.
◆ The Backup Status dialog box appears and displays a Backup Task List that is 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, an e-mail is sent and the event is posted to the Windows Application event log.</td></tr>
</table>

Step	Action
To schedule the backup for later:	
11	If you want to schedule the backup for later, do the following: **a.** Select the Schedule 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. For more information, see “Scheduling a backup job or a database verification job” on page 362.
12	Under “Select the Scheduling Service to Create Job," select either SQL Server Agent or Windows Scheduled Tasks. For more information, see “Scheduling a backup job or a database verification job” on page 362.
13	Click OK. If your backup job is not scheduled, you are taken to the Microsoft SQL Server Management Studio for scheduling. You cannot use the option Windows Scheduled Tasks if your backup job is not scheduled.
14	After the job is scheduled, SnapManager takes you out of the Backup wizard.

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 Verify option. See the following topics for more information:

- "Information you need to specify for a transaction log backup" on page 190
- "Transaction log backup using the Backup and Verify option" on page 193
- "Transaction log backup using the Backup wizard" on page 196

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 backup transaction logs as a follow-on task to a successful full database backup see "Creating a full database backup using SnapManager" on page 176.

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, ensure that a full database backup exists for that database and specify the following information in either the Backup and Verification option or in the Backup wizard:

- Which databases you want to backup

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

  - If the other LUN stores only a single database, you can clear or reselect that database individually.
  - If the other LUN houses multiple databases, you must clear 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* includes 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 contains 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.
- When backup is scheduled for future, existing backups are not deleted unless some of the databases in the server instance are first moved to the local disk.

  If the databases on the LUNs are moved to the local disk later, the scheduled backup operation skips backup deletion. The backup is recreated and rescheduled so that for future operations, backups are deleted first.

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.

- The operation asks whether you want to automatically delete the oldest transaction log backups (recommended 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 168.
- The operation asks 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.
- 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 complete

  For more information about this option, see "Replicating Backups to Mirrored Volumes" on page 267.
- The operation asks 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 a transaction-log-only backup. These settings enable tasks that can be performed after the transaction log backup finishes successfully.

The Backup settings help you determine the following:

- Whether you want to create a backup set of the LUN that contains the SnapInfo directory.
- If you create a backup set of the SnapInfo directory after the backup is complete: whether you also want to delete the oldest SnapInfo directory Snapshot copies and retain only a certain number of the most recent backup sets.
- Whether you want to truncate the transaction log itself. Selecting this option enables you to manage the size of the transaction log.

The preceding options are configured using the Transaction Log Backup option of the SnapManager for SQL Server-Backup dialog box, described in "Configuring the profile of a transaction log backup" on page 429.

The SnapManager for SQL Server-Backup dialog box can be accessed from the Backup and Verify option, as well as from the Backup wizard.

- From within the Backup wizard, you can open the Backup Settings dialog box by selecting the Backup Settings button in the Advanced Backup Options screen.
- If you are using the Backup and Verify option, you can open the Backup Settings dialog box from the Actions pane.

For more details, see "Configuring the profile of a transaction log backup" on page 429.

Transaction log backup using the Backup and Verify option

To start or schedule a backup of only transaction logs using the Backup and Verify option, complete the following steps. If you want to automatically backup transaction logs after a full database backup is complete, see "Creating a full database backup using SnapManager" on page 176 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 190.

Step	Action
1	In the SnapManager console root, click the Backup and Verify option.
2	In the Results pane, select the databases you want to backup. 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 backup" under "Job-specific parameters" on page 190. When you select databases at the SQL Server instance level, SnapManager reports 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 190.
3	In the SnapManager for SQL Server-Backup dialog box, select the "Transaction log backup" option. **Result:** SnapManager 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 log 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.

Step	Action
6	If you want to automatically delete transaction log backups older than a specified number of days, select the "Delete log backups in older than" option and select the number of days. Selecting this option enables you to manage the number transaction log backups.
7	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 prompts you for the details when you are ready to start or schedule the backup operation. For more information, see "Running a script from a UNC path on a Windows Server 2003 system" on page 360.
Back up now or schedule for later	
8	You can either run the backup now or schedule it for later. Click one of the following buttons: ◆ Backup Now... ◆ Schedule...
9	If the Run Command After Operation dialog box appears, specify the command, and then Click OK to close the dialog box.
To complete this procedure by scheduling the backup:	
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 362.

Step	Action
To complete this procedure by starting the backup:	
11	If the Backup Status dialog box appears, do the following: **a.** Read the items displayed in the Backup Task List. This list is 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 complete. ◆ 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 190.

Step	Action
1	In the SnapManager for SQL Server console root, click the SnapManager Backup Wizard option in the Actions pane. **Result:** The SnapManager Backup Wizard starts and displays the Welcome screen.
Welcome	
2	Click Next. **Result:** The Databases to Backup or Verify screen appears.
Databases to Backup or Verify	
3	In the left panel, click to select the databases you want to backup, and 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 backup" under "Job-specific parameters" on page 190. When you select databases at the SQL Server instance level, SnapManager reports 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 190. **Result:** The Backup or Verify Databases and Transaction Logs screen appears.

Step	Action
Back up or Verify Databases and Transaction Logs	
4	Select the Backup Databases and Transaction logs option, and then click Next. **Result:** The SQL Server backup type screen appears.
SQL Server 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. ◆ 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 190.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Backup Finish</td></tr>
<tr><td>7</td><td>The Backup Finish 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 10.</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, Click Finish.

Result: The Backup Status screen appears and displays the backup settings you have selected.</td></tr>
<tr><td>9</td><td>In the Backup Status window, click Start Now.

Result: The backup is performed.
◆ The Backup Status dialog box appears and displays a Backup Task List that is 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, an e-mail is sent and the event is posted to the Windows Application event log.</td></tr>
</table>

Step	Action
To schedule the backup for later:	
10	If you want to schedule the backup for later, do the following: **a.** Select the Schedule 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. For more information, see "Scheduling a backup job or a database verification job" on page 362.
11	Under "Select the Scheduling Service to Create Job," select either SQL Server Agent or Windows Scheduled Tasks. For more information, see "Scheduling a backup job or a database verification job" on page 362.
12	Click OK. If your backup job is not scheduled, you are taken to the Microsoft SQL Server Management Studio for scheduling.
13	After the job is scheduled, SnapManager takes you out of the Backup wizard.

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 12, "Managing SnapManager Operational Reports," on page 263. You can also review the following list of common backup failures:

- "SnapInfo directory being accessed" on page 200
- "SnapInfo directory out of space" on page 200
- "Data does not match" on page 200
- "Backup set already exists" on page 201
- "A database not in valid configuration was not backed up" on page 201
- "[DBMSLPCN] ConnectionRead (WrapperRead())" on page 201

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, ensure that enough space remains in the volume for backup set creation, so that SnapManager can continue to function correctly.

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 refresh your SnapManager view:

- Press F5 on your keyboard.
- From the SnapManager console root, select Action > Refresh SQL Server Configuration.
- Restart SnapManager.

Backup set 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 begin a SnapManager backup, the backup can fail. You can avoid this problem by making sure that SnapMirror replications have enough time to finish before you begin 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 SnapMgrService.exe is still running.

  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.

A database not in valid configuration was not backed up

If a SnapManager backup operation attempts to backup a 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 backup 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 27.

Performing database verification using SnapManager

If you created a full database backup set 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 202
- "Database verification using the Backup and Verify option" on page 205
- "Database verification using the Backup wizard" on page 208

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 fails.

You can run a scheduled verification job that was created using SnapManager installed on a Windows 2003 system, on a remote Windows 2008 system. To reschedule the job, use SnapManager installed on the Windows 2008 system and not the Windows 2003 system.

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 Verify option:

- The databases for which you want to verify any unverified backup sets when you select one database, SnapManager automatically selects all other databases that are located on the same Data ONTAP storage volume.

  You can clear 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 clear or reselect that database individually.
  - If the other LUN houses multiple databases, you must clear or reselect those databases as a group.
- Within the selected databases, the backup management groups for which you want to verify any unverified backup sets

  For more information, see "Using backup management groups in backup and verification" on page 213.

- For the selected databases and backup management groups, the number of unverified backup sets 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.

- 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 complete

  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 267.

- 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 option of the Verification Settings dialog box, described in "Selecting the database verification server" on page 421. If you specify a remote verification server, be sure it is set up properly, as described in "Requirements for a remote verification server" on page 37.

Note

Whether you run database verification either remotely or locally, the verification is performed on a LUN backed by a backup set. If you make a backup set of the same volume while a LUN backed by backup set 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 222.

- Which DBCC options are used to verify database backup Snapshot copies

  This is configured using the DBCC Options option of the Verification Settings dialog box, described in "Selecting DBCC options" on page 423.

The Verification Settings dialog box can be accessed from the Backup and Verify option, as well as from the Backup wizard.

- If you are using the Backup and Verify option, you can open the Verification Settings dialog box by selecting "Verify most recent unverified Snapshot backups only".
- 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 Verify option

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 a backup set. If you make a backup set of the same volume while a LUN backed by backup set 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 222.

Step	Action
1	On the production SQL server, click the Backup and Verify option.
2	Select "Verify most recent unverified Snapshot backups only."
3	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 the databases contained in the most recent backups have already been verified, then SnapManager verifies the databases in the two previous backups.
4	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.
5	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 prompts you for the details when you are ready to start or schedule the verification operation. For more information, see "Running a script from a UNC path on a Windows Server 2003 system" on page 360.

Step	Action
6	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	
7	You can either run the verification now or schedule it for later. Click one of the following buttons: ◆ Verify Now... ◆ Schedule...
8	If the Run Command After Operation dialog box appears, specify the command and then Click OK to close the dialog box.
To complete this procedure by scheduling the verification	
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 362.

Step	Action
To complete this procedure by starting the verification:	
10	If the Backup Status dialog box appears, do the following: **a.** Read the items displayed in the Backup Task List. This list is 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 complete. ◆ 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 202.

Step	Action
1	In the SnapManager for SQL Server console root, click Backup Wizard. **Result:** The SnapManager Backup Wizard starts and displays the Welcome screen.
Welcome	
2	Click Next. **Result:** The Databases to Backup or Verify screen appears.
Databases to Backup 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 backup" under "Job-specific parameters" on page 202.
4	Click Next. **Result:** The Backup or Verify Databases and Transaction Logs screen appears.

Step	Action
Backup or Verify Databases and Transaction Logs	
5	Specify the number of database backup Snapshot copies you want to verify: **a.** Select the "Verify Database and transaction logs 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 "Select the backup management group for this backup" screen appears.
Select the backup management group for this backup	
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 ◆ 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 202.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td colspan="3">Backup Finish</td></tr>
<tr><td>7</td><td colspan="2">The Backup Finish screen prompts you to choose whether you want the operation to be performed now or scheduled for a later time.</td></tr>
<tr><td></td><td>If you want to...</td><td>Then...</td></tr>
<tr><td></td><td>Run the database verification now</td><td>Go to Step 8.</td></tr>
<tr><td></td><td>Schedule the database verification for later</td><td>Go to Step 11.</td></tr>
<tr><td colspan="3">To run the database verification now:</td></tr>
<tr><td>8</td><td colspan="2">If you want to run the database verification immediately, click Verify.</td></tr>
<tr><td>9</td><td colspan="2">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>
<tr><td>10</td><td colspan="2">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 is 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.</td></tr>
</table>

Step	Action
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 362. **Result:** The Select the Scheduling Service screen appears.
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 362. **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.

Step	Action
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 backup set names and SnapInfo directory backup set 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 213
- "How backup management groups are used" on page 214
- "Changing the backup management group of an existing backup set" on page 215

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 Verify option populates 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 176
- "Full database backup using Backup and Verify" on page 181
- "Full database backup using the Backup wizard" on page 185

Note

The type of backup management group—combined with the backup set naming convention selected *(unique* or *generic)*—affects the name assigned to the backup set. The name of each backup set created during a SnapManager backup operation includes information that identifies the backup set contents. This is described in "SnapManager backup set naming conventions" on page 155.

How backup management groups are used

The primary purpose of backup management group is to facilitate a database backup retention 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 backup labeling conventions that determine the backup set's retention policies.

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 176.

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 202.

Options for an explicit deletion of multiple backup sets: When you explicitly delete multiple backups you can specify that only backups belonging to a certain backup management group can be deleted. The procedural details are included in "Deleting backups" on page 218.

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 can 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 console root, click Restore in the Actions pane.
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`*
3	Right-click 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 backup set ◆ The current management group

Step	Action
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 backup set.
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 sets using SnapManager

You can automatically delete older backup sets by specifying the "Delete full backups in excess of" option and the "Delete full backups older than" option in the SnapManager backup facility. This is the recommended method for managing the number of backup sets stored. For more information, see "Automatic deletion of the oldest backups in a management group" on page 168.

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 sets" on page 217
- "Deleting backups" on page 218
- "Busy Snapshot error prevents deletion of backup set" on page 222
- "Busy Snapshot error prevents deletion of backup set" on page 222
- "Automatically delete backup sets" on page 222

Related topics:

- "Ways to manage the number of backup sets kept online" on page 168
- "Deleting restored Snapshot copies" on page 248

Understanding explicit deletion of backup sets

SnapManager provides three ways for you to explicitly delete backup sets.

If you want to delete...	Then use this method...
A specific full database backup set	In the "Delete backups" dialog box, select the databases, the database component types, and the backup management group (Standard, Daily, Weekly, or All). You can also use the Restore option to delete backup sets.
A specific transaction log backup	In the "Delete backups" dialog box, select the "Log Snapshots only" option.

If you want to delete...	Then use this method...
Snapshot copies of LUNs created during restore	In the “Delete backups” dialog box, select the “Delete snapshot of LUNs created during restore” option.

Each of the explicit deletion methods enables you to view detailed information about your selection before you proceed with the operation.

Deleting backups

You can delete backups for a specified group of databases by choosing which backup sets you want to delete and whether you want to also delete LUN backup sets created during the restore (if applicable).

Information you need to specify to delete backups: An explicit deletion of backups is specified using the following parameters.

- The backup sets you want to delete
- The databases for which you want to delete backups
- The backup set components you want to delete: complete data 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
- The number of days such that the backups older than the given number of days are deleted.
- Whether you want to also delete LUN backup sets created during the restore

Procedure: To delete backups, complete the following steps.

Step	Action
1	In the SnapManager console root, select “Delete Backup” from the Actions pane. **Result:** The “Delete backups” dialog box appears and displays information about the selected backup set, including all backed-up databases contained in the backup set.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>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.</td></tr>
<tr><td>3</td><td>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.</td></tr>
<tr><td>4</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>5</td><td>By default, only backups containing all the selected databases are 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 the selected databases</td><td>Go to Step 8.</td></tr><tr><td>Delete backups containing any one or more of the selected databases</td><td>Go to Step 6.</td></tr></table></td></tr>
<tr><td>6</td><td>Click Advanced.

Result: The Advanced Options dialog box appears.</td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>7</td><td colspan="2">Select the given option and 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>
<tr><td rowspan="4">8</td><td colspan="2">Specify which backup sets you want to delete.</td></tr>
<tr><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Delete the oldest backups</td><td>a. Select the “Delete oldest backup in excess of” option.

b. Specify how many of the newest backups you want to preserve.</td></tr>
<tr><td>Delete backups older than a specified number of days</td><td>a. Select the “Delete backups older than” option.

b. Specify how many days of backup you want to preserve.</td></tr>
<tr><td>Delete all the restore Snapshot copies</td><td colspan="2">Select the “Delete all backups in the specified management group” option.</td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="4">9</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 10.</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>10</td><td colspan="2">Click Delete Preview.

Result: The Delete backups dialog box appears. After a moment, the dialog box displays a count and list of the backups identified for deletion.

If you want to view a report, click Show Report.</td></tr>
<tr><td rowspan="3">11</td><td colspan="2">Based on the list displayed in the Delete backups dialog box, you can cancel the delete operation or proceed with the delete operation.</td></tr>
<tr><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Cancel the operation</td><td>Click Close to close the Delete backups dialog box.</td></tr>
<tr><td>Delete the backups listed in the preview</td><td>Click Delete on the Delete backups dialog box.</td></tr>
</table>

Busy Snapshot error prevents deletion of backup set

If you have a backup of a LUN that is backed by another backup set, you get an error stating that the backup set is busy and cannot be deleted.

Definition of a busy backup set: A backup set is busy if there are any LUNs backed by data in that backup set. The backup set contains data that is used by the LUN. These LUNs can exist either in the active file system or in some other backup set. For more information about how a backup set becomes busy, see the *Block Access Management Guide* for your version of Data ONTAP.

If you attempt to delete a busy backup set: If you begin a backup when a LUN backed by a backup set exists, the result is a backup set that cannot be deleted; if you do attempt to delete the backup set, the following events occur:

- SnapManager displays a busy backup set error message.
- SnapDrive logs event 249 in the Windows application event log.

To check whether you have a busy backup set: There are two ways to determine whether you have a busy backup set:

- View your Snapshot copies in FilerView.
- Use the following storage system 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 backup set: Delete the more recently taken backup; then delete the older backup. For more information about deleting a busy backup set, 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 a backup set 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 172.
- While *archiving from a LUN backed by a backup set*, avoid performing a SnapManager backup.

Automatically delete backup sets

SnapManager can be used to automatically delete backup sets as part of a backup. It can also be used to delete backup sets outside the backup process. SnapManager works with SnapDrive to prevent any accidental deletion of Snapshot copies that are required to keep SnapVault up to date.

Note

If Snapshot copies are directly deleted from the storage system without using SnapManager or SnapDrive, do not delete Snapshot copies needed during SnapVault update.

When using SnapVault to archive backup sets in SMSQL, at least two of the most recent snapshots that were used for the SnapVault updates should be kept online in SnapManager.

Example: Assume that four Snapshots are created every day where the first and last Snapshots are used for SnapVault updates and the two Snapshots in the middle are not used for the updates. When using SnapManager to automatically delete Snapshots based on quantity, at least four Snapshots would need to be left online. The two Snapshot copies taken in the middle of the day can be deleted individually and manually under the SnapManager restore option by right-clicking the backup set name and selecting Delete.

Restoring Databases Using SnapManager 10

About this chapter

This chapter describes how to recover your SQL Server databases from SnapManager backup sets. The following topics are covered:

If you are running SQL Server 2000, SnapManager does not allow you to restore a backup set created using SQL Server 2005 or SQL Server 2008.

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 recovery models

SQL Server recovery models

SnapManager 5.0 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 12
- Your Microsoft SQL Server documentation
- Implications for SnapManager operations

The recovery model of a 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:

- Back up the current active transaction log (if possible).
- Restore the most recent database backup without recovery.
- Restore each transaction log backup since the last restored backup.
- Restore the transaction log backup of the currently active transaction log.

If you want to do a full recovery of the master database, clear 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, manually re-execute the Bulk logged operation. Do this if the transaction log that contains the operation's commit record has not been backed up before restore. Hence, if the bulk logged operation inserts 10 million rows in a database and the database fails before the transaction log was backed up, the restored database will not contain the inserted row.

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 228
- "Destinations for a restore operation" on page 228
- "Cloned database in a backup set" on page 229

Related topics

- "Types of SnapManager restore operations" on page 233
- "Choosing the type of restore operation to perform" on page 236

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 server 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 "Preparing to restore operation from a SnapManager backup set" on page 237.

Restore from unmanaged media: You can also use SnapManager Restore to restore databases from offline archives (Unmanaged media) of backup sets.

For information about creating offline archives of backup sets, see "Archiving SnapManager Backups" on page 325.

For information about recovering SQL Server databases from archives, see "Recovering SQL Server databases using archives" on page 304.

Restore a database residing on multiple LUNs or qtrees: You can restore databases that reside on multiple LUNs or qtrees. The restore operation takes some time to complete, due to the fact that SnapManager takes one qtree at a time serially for the complete database restore operation.

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 different database names: You can restore to a different server instance on the same or different server using different database names.

Clone to an alternate location: 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*. You need to clone the database.

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.

Cloned database in a backup set

The following information applies to databases that have been *cloned* or only *mounted at* temporary, alternate locations using writable Snapshot copy.

Backup set label: A database cloned or mounted at a temporary, alternate location is listed in the Backup and Verify option 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 backup set 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 backup set.

Note

If the storage system has a FlexClone license installed, then a FlexClone is used for verification. In this case, you do not encounter the busy Snapshot condition.

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.

How SnapManager Restore works

How SnapManager Restore works

If the “Create transaction log backup before restore” option is selected, the transaction log is backed up before the restore is performed.

If you are cloning the database using a writable Snapshot copy, a transaction log backup is *not* 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 clear this option, see “Understanding the restore options” on page 430.

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 controlled and executed by SnapDrive for Windows. With this method, each of the databases is restored individually. Depending on the composition of the backup set, a stream-based restore can require additional time and free space on the storage system as compared to an online Snapshot copy restore.

SnapManager uses LUN cloning 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 operation (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 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.

In a volume-wide backup, all the databases that reside on a single volume are backed up concurrently using Snapshot copies. Since the maximum number of databases supported per storage system volume is 35,

Total number of Snapshot copies created= Number of databases / 35

If the database has transaction log backups, SnapManager Restore can apply the transaction log backups (if necessary).

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:

`rstsnap__`*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.

SQL Server cluster group state during a restore: If you are using SnapDrive 6.0.1, SnapManager 5.0 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 57.

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 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
Circled check mark	The databases in this backup have been verified.
Circled question mark	The databases in this backup have not been verified.

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 233
- "Point-in-time restore operation" on page 234

Related topics

- "Understanding SnapManager Restore" on page 228
- "Choosing the type of restore operation to perform" on page 236

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:

The last active transaction log is automatically backed up.

The databases are restored from the full database sets you select.

All the transaction logs that were not committed to the databases, including transaction logs from the *backup sets*, from the time the backup set was created up to the most current time, are played forward and applied to the databases (if selected).

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 236.

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, use SnapManager only to back up your SQL Server database transaction log files.

Example: 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 lunch time failed verification, so you decide to restore from the Tuesday lunch time backup. If the 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 (if the transaction logs were backed up).

Point-in-time restore operation

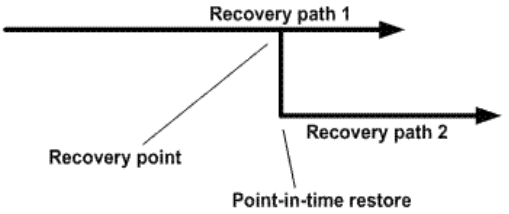
In a point-in-time restore, databases are restored only to a point-in-time from the past. A point-in-time restore occurs in two restore scenarios:

- The database is restored to a given time in a backed up transaction log.
- The database is restored and only a subset of backed up transaction logs are applied to it.

Note

When you restore a database to a point in time, it results in a new recovery path.

The following image illustrates the potential problems when a point-in-time restore is performed.



In the image, Recovery path 1 consists of a full backup followed by the number of transaction log backups. The database administrator restores the database to a point in time. New transaction log backups are created after the point-in-time restores which results in Recovery path 2. The new transaction log backups are created without creating a new full backup. Due to data corruption or other problems, if you need to restore the current database, you will not be able to restore it because a new full backup was not created. Also, it is not possible to apply the transaction logs created in Recovery path 2 to the full backup belonging to Recovery path 1.

Note

Ensure that you always create a full backup after restoring a database to a point in time.

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, or to recover from an accidental database, or table deletion..

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:

- Restore the full database backup taken at midnight and accept the loss of the database changes made afterward.
- Restore the full database backup and apply all the transaction log backups until 9:45 a.m.
- 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.

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 236
- "Requirements and limitations" on page 236
- "Performing a restore operation" on page 237

Related topics

- "Understanding SnapManager Restore" on page 228
- "Types of SnapManager restore operations" on page 233

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 point-in-time restore operation will be available for both up-to-the-minute and point-in-time restore operations.

Performing a restore operation

Performing a restore operation

You can restore SQL Server databases from a SnapManager backup in two ways: using the SnapManager Restore option or using the SnapManager Restore Wizard. See the following topics for more information:

- "Restoring using the SnapManager Restore option" on page 241
- "Restoring using the SnapManager Restore Wizard" on page 245

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.
- 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 console root and select Options > Restore Settings.
- If you restore multiple databases to the same SQL server instance, ensure that you do not assign the same target database name for multiple databases.

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.
- 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.
- A lower version of SQL server cannot restore databases that were created in a higher version of SQL server, but databases created by a lower version of SQL server can be restored by a higher version of SQL Server.

  For example, databases created in SQL Server 2008 cannot be restored by SQL Server 2005, but databases created by SQL Server 2000 can be restored by SQL Server 2005.

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.

Each time you restore a SnapManager backup set, you must specify the following information in either the SnapManager Restore option 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. You should 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.

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 a 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 233 and "Choosing the type of restore operation to perform" on page 236.

You cannot restore multiple databases with different restore options in a single restore operation.

Restore location:

- Original database (the default setting)
- The transaction log backup sets to be restored

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 databases" on page 449.

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 databases" on page 449.

Restoring data backed up using filestreaming

If you are restoring or verifying database that was backed up using filestreaming, ensure that the filestreaming option is enabled at SQL Server instance level. For more information on how to enable Filestream option, refer to SQL Server online documentation.

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 option of the Verification Settings dialog box. See "Selecting the database verification server" on page 421.
- Which DBCC options are used to verify database backup sets

  This is configured using the DBCC Options option of the Verification Settings dialog box. See "Selecting DBCC options" on page 423.
- The Verification Settings dialog box can be accessed from the Restore wizard.

Restore settings

The following restore settings determine how SnapManager is to restore database backup sets:

- 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

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 431.

Using the Find Backups Wizard

You can restore backups that were created previously using the Find Backups Wizard. Follow these steps to restore backups created previously.

Step	Action
1	Click Restore in the Scope pane.
2	Click Find Backups in the Actions pane. **Result:** The SnapManager for SQL Server Find Backups Wizard starts.
3	Follow the steps as instructed in the wizard and click Finish.

Using this wizard, you can restore backups that were created on the same SQL server, restore from unmanaged media or restore backups that were created on a different server by selecting the relevant option in the wizard. You need to enter the SnapInfo directory path if you want to restore from unmanaged media or restore backups that were created on a different server.

Restoring using the SnapManager Restore option

To restore a SQL Server database from a backup set using SnapManager Restore, complete the following steps.

Step	Action
1	Review the list in "Preparing to restore operation from a SnapManager backup set" on page 237.
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.
4	In the SnapManager console root, click the SQL server you want to perform the restore operation on.
5	In the Scope pane, click Restore. **Result:** You can now navigate the backup sets.
Specify the source	
6	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 SnapManager Restore" on page 311.
7	In the Results panel, double-click to select the backup set from which you want to restore.
8	In the Actions panel, click Restore. **Result:** The SnapManager for SQL Server-Restore window appears.

Step	Action
9	If you want to restore as a database with a different name than the original database, follow these steps: **a.** Click the tab marked "..." next to "Restore as Database". **b.** 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. **Note** As soon as you have completed viewing the data, detach the database and dismount the Snapshot copy LUNs.
10	Click the tab marked "..." next to "Restore to Server (instance)".
11	Select or enter the server name that you want the database to be restored to.
12	Choose the Connection by selecting the "Use Windows Authentication" or "Use SQL Server Authentication" radio button.
13	Click OK to apply your change and close the dialog box.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Specify the restore type</td></tr>
<tr><td>14</td><td>If you want to perform a point-in-time restore, do the following:
a. Click the tab marked "..." next to "Point-in-Time Restore"
The Point-in-Time 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.
Note
A point-in-time restore will halt the restoration of transaction log entries that were recorded after the specified date and time.</td></tr>
<tr><td>15</td><td>To coordinate backup and restore processes, click Advanced Options. In the Job Control Option pane, select one of the following options.
<table>
<tr><th>If...</th><th>Then...</th></tr>
<tr><td>You want to cancel the current backup operation that is in progress</td><td>Select the "Cancel conflicting backup that is in progress" option.
Result: SnapManager pauses all the active scheduled backup jobs on the current SQL server, or on all nodes in the cluster environment, and cancels the current backup before performing the restore operation. Upon completing the restore operation, SnapManager enables the paused scheduled backup jobs. All the other inactive jobs will not get changed.</td></tr>
<tr><td>You want to abort the restore operation</td><td>Select the "Abort restore if conflicting operation is running" option.
Result: The restore operation is aborted.</td></tr>
</table></td></tr>
</table>

Step	Action
16	To start the restore operation, click Restore. **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.
17	After all the restore tasks are finished, click OK. **Result:** Your restore is complete and your SQL Server computer comes back online.
18	After the restore is complete, you can optionally perform a SnapManager backup and verification to verify that your restored database is free of physical-level corruption.

Other restore options in the Actions pane

You can also change the management group of the database to be restored using the option “Change Management Group.” You can also mount Snapshots, run the DBCC functionality, and attach a copy of databases to Snapshots using the option “Mount Attach Db...”

Note

Detach the copy of the database from the Snapshots and dismount the Snapshots manually using SnapManager Backup after you have completed your tasks to avoid creating a busy Snapshot copy.

Restoring using the SnapManager Restore Wizard

To restore a SQL Server database from a backup set using the SnapManager Restore Wizard, complete the following steps.

Step	Action
1	Review the list in "Preparing to restore operation from a SnapManager backup set" on page 237.
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.
4	To launch the SnapManager Restore Wizard, select the server you want to restore to in the Scope pane.
5	Select "Restore Wizard" from the Actions pane. **Result:** The Restore Wizard appears and displays the Welcome screen.
Welcome	
6	Click Next. **Result:** The SnapManager for SQL Server Restore screen appears.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td colspan="3">SnapManager for SQL Server Restore</td></tr>
<tr><td>7</td><td colspan="2">By default, SnapManager restores from backups that were created on the same server that you run the Restore wizard on.</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You want to restore from backups that were created on the same SQL server</td><td>Select “Restore SnapManager backups that were created on the same SQL server”.

The “Backup Set” screen appears. See Step 8.</td></tr>
<tr><td></td><td>You want to restore from backups that were created on a different SQL Server</td><td>Select “Restore backup created on a different server”.

Follow the procedure described in “Restoring from other SQL Server backups using the SnapManager Restore wizard” on page 316.</td></tr>
<tr><td></td><td>You want to restore from an unmanaged media</td><td>Select “Restore from Unmanaged Media”.</td></tr>
<tr><td colspan="3">Backup Set</td></tr>
<tr><td>8</td><td colspan="2">Double-click to select the backup under the database you want to restore.

Click Next.

Follow the procedure described in “Restoring databases from other SQL Server backups” on page 311.</td></tr>
<tr><td>9</td><td colspan="2">Follow the instructions in the Restore Wizard as you proceed.</td></tr>
</table>

Step	Action
Completing the Restore Wizard	
10	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	
11	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, an e-mail is sent to the specified address. All events are posted to the Windows event log, even if notification is not enabled.
12	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.
13	After the restore is complete, you can optionally perform a SnapManager backup and verification to verify that your restored database is free of physical-level corruption.

Deleting restored Snapshot copies

Deleting restored Snapshot copies

To explicitly delete the oldest LUN Snapshot copies created during previous restore operations, complete the following steps.

Step	Action
1	In the SnapManager Actions pane, click Delete Backup. **Result:** The Delete backups dialog box appears.
2	Select “Delete Snapshot of LUNs created during restore” option. For more information, see “Explicitly deleting backup sets using SnapManager” on page 217.

Restoring replicated publisher and subscriber databases

Restoring replicated publisher and subscriber databases

If you are restoring replicated publisher and subscriber databases, follow these steps:

Step	Action
1	Perform the backup operation on the distribution and replicated database.
2	Restore the following database strictly in the given order: **a.** Distribution database **b.** Publisher database **c.** Subscriber database **Note** If you do not restore the distribution database first, the replication settings are not maintained and you will have to restart the replication.
3	While restoring replicated databases, stop the running SQL Agent.
4	Take the publisher and subscriber database offline.
5	In the Action pane, click Restore Settings> Restore SQL database replication settings.
6	Select the options “Retain SQL database replication settings” and “Restore database even if existing databases are online”.
7	If you have multiple replication sets, restore the most recent distribution database to maintain the replication settings for all of the other replicated databases.
8	Reinitialize the restored publisher or subscriber databases because they are out of sync with the latest distribution database.

Cloning Databases 11

About this chapter

This chapter describes how to clone SQL Server databases in production and in a backup set. It contains the following topics:

Understanding Database cloning

What Database cloning is

Database cloning is the process of creating a point in time copy of a production database or its backup set.

Cloned databases can be used for multiple purposes:

- During application development cycles for testing functionality that has to be implemented using the current database structure and content.
- By data extraction and manipulation tools for populating data warehouses.
- Recovering data that was mistakenly deleted or changed.

Database cloning feature enables you to clone all databases in a LUN simultaneously, or select specific databases out of many. You can either rename the cloned database or leave the default value provided. You can select the SQL Server instance either from a host on which the database resides or from a remote host. Clone to a virtual SQL Server instance is not supported, only local instances are supported for database cloning operations.

Note

The remote host must be connected to the storage system containing the database files.

To optimize disk space usage, it is highly recommended that you delete cloned databases that are no longer relevant.

Completion of a current database cloning operation generates two reports: a backup report and a restore report.

Types of clone operations performed using SnapManager

Tasks performed by the Clone wizard

SnapManager contains a Clone wizard that provides a convenient interface for performing the following cloning operations:

- Clone database from an existing backup set
- Clone active production databases
- Delete cloned database

Note

Cloning using the Clone wizard provides you with a complete set of cloning options. Cloning using the Actions pane in SnapManager Restore gives you quick cloning with fewer options than the Clone wizard.

Cloning a database that is in production

The clone of a database that is in production is used when a new application or function has to be tested with the latest database content as the final step before the application is taken into production.

A current database that is in production must be selected for cloning. Cloning a current database involves two steps. The first step is creating the backup of the selected database and the second step is to restore the database from the just created backup set. The whole cloning process is managed by the Cloning wizard. Options made visible by the Cloning wizard are similar to options available in the Backup and Restore wizard.

To clone a current database perform the following steps:

Step	Action
1	If you have not already done so, start SnapManager by accessing the Windows Start menu, and selecting Program Files > NetApp > SnapManager for SQL Server. **Result:** The SnapManager for SQL Server console appears.
2	In the Scope pane, double-click SnapManager for SQL Server. **Result:** SnapManager displays the servers running.

Step	Action
3	Click the SQL Server database server that you want to configure. **Result:** SnapManager displays the Status dashboard in the Result pane.
4	In the Actions pane, click Clone Wizard. **Result:** The Clone wizard launches and the Welcome window appears.
5	Click Next. **Result:** SnapManager displays an option for selecting the operation that you want to perform, which is clone database in production.
6	Select the operation you want to perform, and click Next. **Result:** SnapManager displays the option for selecting the databases that you want to clone.
7	If you select the option “Clone Active Production Database”, you get the option “Run Through Clone QuickStart Wizard”. This wizard takes you through the cloning of active production database.
8	Select the databases that you would like to clone and click Next. **Result:** SnapManager displays the option for selecting the backup management group. **Note** The first time you select a database on a LUN, SnapManager automatically selects all other databases on the same LUN. You can then de-select any databases that you do not want to be cloned. When you de-select one of the databases on a LUN, SnapManager displays a window that advises you to clone all databases on a LUN and also asks you to confirm your selection. Click Yes to confirm your selection and No to de-select all databases on the LUN.
9	Continue with the next steps as instructed in the wizard.

Step	Action
10	The wizard takes you to the final option that displays the SnapManager clone task list. Click start now to begin the specified tasks. **Result:** The operation is performed, and each item in the Clone Task List is checked off as the task is complete. A message appears indicating the successful completion of the cloning operation.
11	Click Close to close the Clone Status dialog box.

Cloning a database in a backup set

Cloning the backup of a database is probably the most commonly used cloning feature. The cloned database can serve as a baseline for developing new applications, or to isolate application errors that occur in the production environment. It could also be used for recovery from soft database errors.

Perform the following steps to clone a database in a backup set.

Step	Action
1	If you have not already done so, start SnapManager for SQL Server by accessing the Windows Start menu, and selecting Program Files > NetApp > SnapManager for SQL Server. **Result:** The SnapManager for SQL Server console appears.
2	In the Scope pane, double-click SnapManager for SQL Server. **Result:** SnapManager displays the SQL Server database servers running.
3	Click the server that you want to configure. **Result:** SnapManager displays the Status dashboard in the Result pane.
4	In the Actions pane, click Clone Wizard. **Result:** The Clone wizard launches and the Welcome window appears.

Step	Action
5	Click Next. **Result:** SnapManager displays an option for cloning database in a backup set.
6	Select the operation you want to perform, and click Next. **Result:** SnapManager displays the option for selecting the databases you want to clone.
7	Specify the Server where you want the clone database to reside. You can also select the server by clicking on Browse.
8	SnapManager displays the list of databases to be cloned. By default, SnapManager provides the same name to the clone as the original database. It is recommended that you rename the cloned database. Click Next. **Result:** SnapManager displays the option for specifying the SQL Server instance on which the clone will reside.
9	Specify the name you want to clone the database as and the name of the clone server instance in the upcoming screens.
10	Click the Clone Settings button. **Result:** The Restore Settings dialog box appears.
11	Select the restore settings you want to apply to the cloned database.
12	Click Next. **Result:** The “Database state after restore” screen appears.
13	Select the state of the database you want after restore. If you select "Leave the database in read-only mode and available for restoring additional transaction logs”, the “Undo file directory” option activates. **Note** The default path for the SnapInfo directory in the “Undo file directory” option is that of the source host.

Step	Action
14	Specify a valid path for the SnapInfo directory in the destination host.
15	Click Next.
16	Click Finish. **Result:** The Clone Status window is displayed that shows the Clone task list and the Clone Reports.
17	Click Start Now to start cloning. The operation is performed, and each item in the Clone Task List is checked off as the task is complete. A message appears indicating the successful completion of the cloning operation.

Cloning using the Clone option in SnapManager Restore

To clone an SQL server database from a backup set using SnapManager Restore, complete the following steps.

Step	Action
1	Review the list in "Understanding SnapManager Restore" on page 228.
2	Ensure 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.
4	In the SnapManager console root, click the SQL server you want to perform the restore operation on.
5	In the Scope pane, click Restore. **Result:** it enables you to navigate the backup sets.
6	In the Results panel, double-click to select the backup set that you want to clone.

Step	Action
7	In the Actions panel, click Clone. **Result:** The SnapManager for SQL Server-Clone window appears.
Specify the destination	
8	If you want to clone a database on a server other than the original server, follow these steps: **1.** Click the option marked "..." next to "Clone to Server (instance)". **2.** The "Select SQL Server Agent" dialog box appears. **3.** Select the SQL Server to which you want to restore the database. **4.** Choose the Connection by selecting the "Use Windows Authentication" or "Use SQL Server Authentication" option. **5.** Click OK to apply your change and close the dialog box.
9	If you want to clone a database with a different name, follow these steps: **1.** Click the option marked "..." next to "Clone as Database". The "Individual Database Restore As..." dialog box appears. **2.** 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. **3.** Click OK to apply your change and close the dialog box. **Note** When you have completed viewing the data, detach the database and dismount the Snapshot copy LUNs.

Step	Action
Specify the restore type	
10	If you want to perform a *point-in-time* restore, do the following: **a.** Select the Point in Time Restore option. The Point-in-Time 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 the option marked "..." next to "Point in Time Restore". **Note** A point-in-time restore will halt the restoration of transaction log entries that were recorded after the specified date and time.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>11</td><td colspan="2">To coordinate backup and restore processes, click Advanced Options. In the Job Control Option pane, select one of the following options.</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You want to cancel the current backup operation that is in progress</td><td>Select the “Cancel conflicting backup that is in progress” option.

Result: SnapManager pauses all the active scheduled backup jobs on the current SQL server, or on all nodes in the cluster environment, and cancels the current backup before performing the restore operation. On completing the restore operation, SnapManager enables the paused scheduled backup jobs. All the other inactive jobs will not get changed.</td></tr>
<tr><td></td><td>You want to abort the restore operation</td><td>Select the “Abort restore if conflicting operation is running” option.

Result: The restore operation is aborted.</td></tr>
<tr><td>12</td><td colspan="2">To start the restore operation, click Clone.

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, an e-mail is sent to the specified address. Events are posted to the Windows event log even if notification is not enabled.</td></tr>
</table>

Step	Action
13	After all the restore tasks are finished, click OK. **Result:** Your restore is complete and your SQL Server computer comes back online.
14	After the restore is complete, you can optionally perform a SnapManager backup and verification to verify that your restored database is free of physical-level corruption.

Deleting cloned databases

A cloned database that has outlived its purpose may be deleted for conserving disk space. Deleting the cloned database implies disconnecting the LUNs.

Perform the following steps to delete a cloned database.

Step	Action
1	If you have not already done so, start SnapManager for SQL Server by accessing the Windows Start menu, and selecting Program Files > NetApp > SnapManager for SQL Server. **Result:** The SnapManager for SQL Server console appears.
2	In the Scope pane, double-click SnapManager for SQL Server. **Result:** SnapManager displays the SQL Server database servers running.
3	Click the SQL Server database server that you want to configure. **Result:** SnapManager displays the Status dashboard in the Result pane.
4	In the Actions pane, click Clone Wizard. **Result:** The Clone wizard launches and the Welcome window appears.
5	Click Next. **Result:** SnapManager displays an option for selecting the operation that you want to perform.

Step	Action
6	Select the operation you want to perform, and click Next. **Result:** SnapManager displays the Database to clone window listing the available cloned databases. Select the cloned databases that you want to delete.
7	In the Delete clone summary screen, verify the settings selected in the previous steps and click Finish.
8	This takes you to the final option that displays the Delete clone task list. Click start now to begin the specified tasks. **Result:** The operation is performed, and each item in the Clone Task List is checked off as the task is complete. A message appears indicating the successful completion of the delete clone operation.
9	Click Close to close the Clone Status dialog box.

Managing SnapManager Operational Reports 12

This section describes the reports that SnapManager automatically creates for every operation performed. The following topics are covered:

- "Understanding the SnapManager Reports option" on page 264
- "Managing reports" on page 266

Related topics:

- "Uninstalling SnapManager" on page 70
- "SnapManager report directory options" on page 440

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 option

Understanding the SnapManager Reports option

Use the SnapManager Reports option in the Scope pane 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 option 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 time stamp 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.

Debug: Contains a log file for every debug operation you carry out in SnapManager.

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 option can contain more folders.

Viewing a SnapManager report

Managing reports

Viewing reports

To view a SnapManager report, complete the following steps.

Step	Action
1	In the SnapManager console root, click the SnapManager Reports option.
2	In the navigation panel, click to expand the appropriate reports folder and select the report you want to display in the display panel.

Managing reports

To manage your reports, perform the actions listed in the table below:

If you want to...	Then...
Refresh a report that you are viewing	Click “Refresh” in the Actions pane.
Delete a specific report	Select the report and click “Delete” in the Actions pane.
Delete all reports of a folder	Select the folder in the Results pane and click “Delete All” in the Actions pane.
Open the report in a new window	Select the report and click “New window” in the Actions pane.
Open the report in Notepad	Select the report and click “Open in Notepad” in the Actions pane.
Print the report	Select the report and click “Print” in the Actions pane.
Preview the print layout of the report	Select the report and click “Print Preview” in the Actions pane.
Find a word in a report	Select the report and click “Find in Report” in the Actions pane.

Replicating Backups to Mirrored Volumes 13

About this chapter

This chapter 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:

Related topics:

- "Backing Up Databases Using SnapManager" on page 173
- "Performing Disaster Recovery with SnapManager" on page 285

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

About this section

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 268
- "How SnapManager uses SnapMirror" on page 270
- "Process overview" on page 271

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. SnapMirror can replicate a source volume to a destination volume on the same storage system or to a different storage system. When you use SnapMirror to replicate volumes from one storage system to another, the destination storage system can be in a different geographical location. This ability to duplicate data in different locations is a key component of a sound disaster recovery plan.

The data stored in a mirror on a destination volume can be accessed through SnapDrive. Because the mirror is volume wide, Snapshot copies of other datasets on the source volume are mirrored also.

Note

Because a mirror is volume wide, it includes Snapshot copies of any other datasets 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 non-production 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.

Attention

As SnapManager uses SnapMirror in asynchronous mode, any disk writes that occurred on the source volume after the most recent SnapMirror replication update are not available if a catastrophic failure occurs before the next update. This is because they were not replicated to the SnapMirror destination volume.

How SnapManager uses SnapMirror

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 begin 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 273, 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 on the completion of a SnapManager backup operation.

A subsequent section, "Scheduling SnapManager backups with SnapMirror replication" on page 277, describes how to do this plus other steps in the process.

Requirements for using SnapMirror with SnapManager

To use SnapMirror with SnapManager, you must have previously configured SnapMirror on both the source volume to be replicated and its destination volumes. How to complete these tasks is explained in your SnapDrive documentation. Your configuration must satisfy the following requirements:

- There must be one or more SnapMirror source volumes hosting LUNs.
- There must be one or more SnapMirror destination volumes for each source volume.
- The size of the destination volumes must be equal to or greater than the size of the source volume.
- SnapMirror licenses must be enabled on both the source and destination storage systems.
- You must manually configure and initialize the SnapMirror replication between source and destination volumes.
- You must disable the SnapMirror replication schedule on your storage system.
- You must configure the SnapMirror replication as asynchronous.

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 2.0, SnapManager 2.1 for Microsoft SQL Server, and SnapManager 5.0 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

About this section

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 273
- "Supplemental replication using rolling Snapshot copies" on page 273
- "Supplemental mirroring of the transaction logs" on page 275

Goal: More frequent mirror updates with minimal Snapshot overhead

As described in "SnapMirror replication for SnapManager is asynchronous" on page 270, 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 databases per LUN" on page 168.

As an alternative to increasing the frequency of SnapManager backups, you can use SnapDrive to initiate additional, more frequent SnapMirror replication updates. In order to minimize exposure to data-loss, it is advisable to keep the transaction log size small and take more frequent transaction log backups.

Supplemental replication using rolling Snapshot copies

When you use SnapDrive to begin 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, in addition to 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 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 begin 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 be restored from a backup set.
- If the SQL Server on the primary site is available, then SnapManager will request that SQL Server instance to backup 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.

Backup and replication schedule: Schedule your transaction log backups to be more frequent than the full backup job. For example, if you have scheduled full backups at an interval of four hours, you can schedule transaction log backups at an interval of 15 minutes by replicating the SnapInfo directory. This ensures that you do not lose any modifications to your data files.

In case of a disaster, bring the data file backup, the transaction log backup and SnapInfo directory online essentially. Now perform the normal restore procedure.

Scheduling SnapManager backups with SnapMirror replication

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 176
- "Performing database verification using SnapManager" on page 202
- "Scheduling a backup job or a database verification job" on page 362

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 non-schedule 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 Backup and Verify 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 Backup and Verify, click Schedule instead of Backup Now or Verify Now. ❖ If you are using the Backup wizard, click Schedule.

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 275. For more information about using the Schedule dialog box in general, see "Scheduling a backup job or a database verification job" on page 362

Integrity verification on the SnapMirror destination volume

Choosing the SnapMirror destination volumes

SnapManager enables you to verify the SQL Server databases stored on the LUNs of the destination SnapMirror volumes. When verifying the integrity of databases on a destination volume, SnapManager automatically detects the existing SnapMirror relationships in the SQL Server volumes and selects the available SnapMirror relationship for the selected destination volume.

The "Choose SnapMirror Destination Volumes for Integrity Verification" window shows the relationship between the source and the destination SnapMirror volumes. Each SnapMirror volume is displayed as a tree showing the relationship among the volume, the LUN, and the databases contained in it. For each source volume, there is a list of destination volumes, and each destination volume displays a SnapMirror state and a FlexClone state. You can select a SnapMirror destination volume for each SnapMirror source volume for which you want to verify integrity.

To select the destination volume for each SnapMirror relationship, complete the following steps.

Step	Action
1	Click Backup Verification Settings in the Actions pane. **Result:** The Verification Settings dialog box appears.
2	Click the Verification Server tab.
3	In the Verification Server tab, click "Verification on destination volumes" button. **Result:** SnapManager displays the Choose SnapMirror Destination Volumes for Integrity Verification window. If the SnapMirrored volume is not available, SnapManager displays an appropriate error message.

Step	Action
4	Select the destination volume for each SnapMirrored volume. By default, SnapManager displays the "Number of Relationships" field. You cannot edit this value. If the destination volume is not in the SnapMirrored state or does not have FlexClone license installed, SnapManager displays an error message when you click Apply. Integrity verification completes for the source verification volume but fails for the destination volume.
5	Click Apply to save the changes. **Result: :** SnapManager saves the settings and makes it available whenever you launch the SnapManager application.
6	Click OK.

Understanding the requirements to run integrity verification

To run integrity verification on destination SnapMirror volumes, ensure that the following system configuration requirements are met:

- A SnapMirror license is enabled on the source volume and a FlexClone license is enabled on the destination volume. SnapManager uses SnapDrive to verify that the required licenses are enabled on the source and destination storage system.
- To run integrity verification on the destination volume, the destination volume must have CIFS shares configured, to be accessible by SnapDrive.
- SnapDrive provides access to the SQL server databases that are stored on the destination volume, and SnapManager performs the integrity verification on backups of those databases.
- Data files and log files should be present on the SnapMirror destination volume.

Note

SnapManager fails the mount operation on a Snapshot copy in the destination volume if the FlexClone license is not enabled on the SnapMirror destination volumes.

Understanding different types of integrity verification

You can run integrity verification on the SnapMirror destination volumes for different SnapManager operations:

- Full database backup verification
- Deferred integrity verification
- Mount Snapshot and Attach Database
- Restore
- Remote verification of full database backup

Full database backup verification: When you run integrity verification on the SnapMirror destination volume, SnapManager performs the following operations:

- Creates a Snapshot copy of the database volumes
- Requests SnapMirror update to replicate the data across destination volumes through SnapDrive

  When the SnapMirror update replicates the backup to the selected destination, SnapManager continuously monitors the SnapMirror update activity through SnapDrive, as follows:

  - SnapDrive provides the SnapMirror update progress information continuously to SnapManager during the update.
  - SnapManager logs the SnapMirror update activity to the backup report at every defined interval.

Note

If the SnapMirror update operation does not have any progress within a defined interval, SnapManager aborts monitoring it and leaves the backup unverified.

- Create a Snapshot copy for the SnapInfo volume
- Mount the available database LUNs in the Snapshot copy on the destination volumes.
- Verifies the integrity of databases and transaction logs from LUNs located in the selected destination volumes
- Dismounts the database LUNs in the Snapshot copy on the destination volumes.
- Updates database integrity verification result to the live backup SnapInfo.
- Request SnapMirror update to replicate the database verification result to the destination volumes.

Deferred integrity verification: For deferred integrity verification (that is, verification at some stage after the backup has been created), SnapManager verifies the SnapMirror state of the destination storage system volume and the existence of a backup Snapshot copy on it. If the Snapshot copy does not exist, SnapManager displays an error message.

When you run a deferred integrity verification on available SnapMirror destination volumes, SnapManager performs the following actions:

If the "SnapMirror update after operation" option is not selected, SnapManager updates the verification results only on the source SnapInfo volumes.

If the "SnapMirror update after operation" option is selected, the following actions occur:

- SnapManager verifies the backup on the selected destination.
- SnapManager updates the verification results to the source SnapInfo volumes.
- SnapMirror replicates the verification results on the source SnapInfo volumes to all the destination SnapInfo volumes.

If the backup is not available on the destination volume, SnapManager fails the mount operation for integrity verification and leaves the backup unverified. In this case, SnapManager does not request the SnapMirror update.

Mount Snapshot and Attach Database: When you run an integrity verification for the Mount Snapshot and Attach Database operation, SnapManager performs the following tasks:

- If "Run DBCC CHECKDEB" option is selected, it performs integrity verification on the available selected destination volumes. If "Leave database attached after DBCC" option is selected, the database from the Snapshot copy in the FlexClone destination volumes remains online and operational.
- Displays an appropriate error message, if a backup is not available on the destination volume.
- Updates the verification results to the source SnapInfo volumes

Note

SnapManager does not verify the backup on the source volume, if an unverified SnapManager backup is not available on the destination volume. If you try to run integrity verification in such a case, SnapManager displays an error message.

Restore: When you run an integrity verification for a restore operation on the destination volume, SnapManager performs the following actions:

- Performs integrity verification on the destination SnapMirror volumes if a backup is available on the destination volume
- Displays an appropriate error message, if a backup is not available on the destination volume
- Updates the verification results to the source Snapinfo volumes
- Logs the additional steps to the Windows Application event log and to the SnapManager restore report

Remote verification: When you run an integrity verification on the destination volume present on the remote destination, SnapManager performs the following actions:

- Creates a backup for the selected database
- Requests SnapMirror Update to replicate the new data to the destination volumes
- Creates a backup for the SnapInfo volume
- Requests SnapMirror Update to replicate the new SnapInfo data to the Destination volumes
- Mounts the database LUNs in the SnapShot copy on the destination volumes
- Performs database integrity verification
- Dismounts the database LUNs in the SnapShot copy on the destination volumes
- Updates database integrity verification result to the live backup SnapInfo
- Requests SnapMirror Update on the SnapInfo volume to replicate the database verification result to the destination volume

Performing Disaster Recovery with SnapManager 14

About this chapter

This chapter describes how to recover from catastrophic loss of a SQL Server in a SnapManager environment. The following topics are covered:

Related topics:

- Chapter 13, "Replicating Backups to Mirrored Volumes," on page 267
- Chapter 15, "Archiving SnapManager Backups," on page 325

Preparing for disaster recovery

About this section

The following topics discuss preparation for disaster recovery:

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
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 For Microsoft SQL Server 2008: ◆ *Microsoft SQL Server Books Online* (installed with the application) ◆ Any related Microsoft documentation
Data ONTAP	*Data ONTAP System Administrator's Guide* for your version of Data ONTAP

For information about	See this document
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

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 a 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 non-production 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 backup or replicate your Windows environment. For more information about backing up your Windows environment, see "Backing up your Windows environment" on page 289.

SQL Server: Back up your SQL Server. For more information about backing up your SQL Server, see "Backing up your SQL Server" on page 289.

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 storage system 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 13, "Replicating Backups to Mirrored Volumes," on page 267.
- Archive the data to physical media, such as tapes, and store that media offsite. For more information, see Chapter 15, "Archiving SnapManager Backups," on page 325.
- Automatically archive SnapManager backups using SnapVault. 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 enough bandwidth. ◆ Mirrors only backup 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 dataset and SnapVault integration with SnapManager	◆ Restoring from a SnapVault archive is faster than archiving from tape. ◆ You can create and restore remote backup and archives. ◆ 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

About this section

The following topics describe backing up your SQL Server environment:

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 backup your Windows environment so that you can restore the same state as part of the recovery process.

To backup 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 backup 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

For Microsoft SQL Server 2008:

- *Microsoft SQL Server Books Online*
- Any related Microsoft documentation

Replicating your SQL Server environment

About this section

The following topics describe replication of your SQL Server environment:

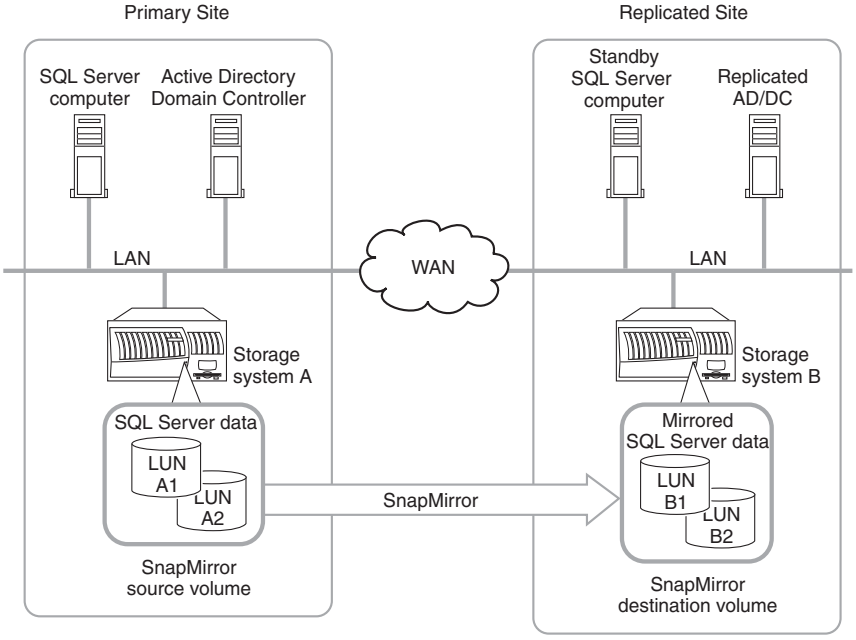
- "Reason to replicate your SQL Server environment" on page 291
- "Example of a replicated site" on page 291

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 storage system is mirrored using SnapMirror to a storage system 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

About this section

The following topics describe the recovery from catastrophic loss of a SQL Server environment that integrates SnapManager and storage systems:

- "SQL Server environment recovery processes" on page 293
- "Rules and guidelines for disaster recovery" on page 293
- "Choosing a recovery procedure" on page 294

SQL Server environment recovery processes

Recovery from catastrophic loss of a 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 "Preparing for disaster recovery" on page 286.

Recovering the SQL Server: This process is performed using the backup application you used to backup 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.
- To be able to perform an up-to-the-minute restore in a SnapMirror configuration, you would need to have incorporated supplemental rolling Snapshots along with more frequent mirroring of transaction logs.

  This is described in "Minimizing your exposure to loss of data" on page 273.

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 296
Your SQL Server computer is online and you want to recover SQL Server databases from archive	"Recovering SQL Server databases using archives" on page 304

If...	Then use...
You are creating or restoring archives at remote storage system	"Dataset and SnapVault Integration" on page 343
Your SQL Server computer has failed or been destroyed	"Recovering a failed SQL Server computer" on page 306
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 309

Recovering SQL Server databases using SnapMirror

About this section

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

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: Use SnapDrive for Windows to connect to the data file backups, transaction log backups, and SnapInfo LUNs and then perform a SnapManager based restore operation for the concerned database.

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 backup 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 233.

- 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 and SQL Server 2008 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 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 replica 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 155. ◆ For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 154 and "Transaction log backup" on page 161. ◆ For information about performing a restore operation, see "Restoring using the SnapManager Restore option" on page 241. The procedure is now complete.
Performing the restore if the transaction log volume was lost	
8	Disable the option to backup the transaction log before performing the restore: **a.** From the SnapManager menu bar, select Options > Restore Settings. **b.** In the Restore Settings dialog box, clear the "Create transaction log backup before restore" option. **c.** Click OK. 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 155. ◆ For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 154 and "Transaction log backup" on page 161. ◆ For information about performing a restore operation, see "Restoring using the SnapManager Restore option" on page 241. 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: Back up the Last Transaction Log When the Master and the Database Files Are Damaged." This article describes how you can backup the currently active transaction log even if the SQL Server 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 backup the transaction log (as described in the Microsoft article), go to the next step in this procedure. For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 154 and "Transaction log backup" on page 161. **Attention** Do not use SQL Server Enterprise Manager or SQL Server Management Studio to backup 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 backup 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 backup 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 155. For general information about performing a restore operation, see "Restoring using the SnapManager Restore option" on page 241.

Recovering SQL Server databases using archives

About this section

The following topics describe recovery of SQL Server databases using archives:

System prerequisites

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 Server: 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 Unmanaged Media" 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.

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 Server registry
  - Windows Server boot files
  - Windows Server 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.	
3	**If...**	**Then...**
	Your backup software was configured to backup SnapDrive	Continue with Step 4.
	Your backup software was not configured to backup 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.

Step	Action	
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.	
5	**If...**	**Then...**
	Your backup software is configured to backup the entire SQL Server computer less the databases	Perform a complete recovery of your SQL Server computer using the same backup application. Databases are later restored with SnapManager.
	You are not backing up the SQL Server computer other than using SnapManager to backup the databases	Reinstall the SQL Server software and apply any necessary service packs.
6	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.	
7	Restore the most recent backup using SnapManager Restore. Do not select the Point-in-Time restore option. See "Restoring using the SnapManager Restore option" on page 241.	
8	Confirm the operation of the SQL Server.	

Recovering both a failed storage system and a failed SQL Server computer

Recovering both a failed storage system and a failed SQL Server computer

If both the storage system and the SQL Server computer fail, usually 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 a 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 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 a 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 Server and load the appropriate service packs. **b.** Install Microsoft SQL Server and load the same service pack that was on the system before the failure. **c.** Install SnapDrive and connect to the same drive letters that your LUNs were connected to before the failure. **d.** Assuming you backed up your system databases, install SnapManager and migrate your system databases to the same LUNs that they were migrated to before the failure.
3	Using SnapManager, recover your SQL Server system databases (master and msdb) from the archived LUN. See “Recovering SQL Server databases using archives” on page 304.
4	Using SnapManager, recover the user databases.

Restoring databases from other SQL Server backups

About this section

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 option or the SnapManager Restore wizard. For detailed information, see one of the following topics:

- "Restoring from other SQL Server backups using SnapManager Restore" on page 311
- "Restoring from other SQL Server backups using the SnapManager Restore wizard" on page 316

Restoring from other SQL Server backups using SnapManager Restore

To use the SnapManager Restore option 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.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td colspan="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.</td></tr>
<tr><td rowspan="3">3</td><td colspan="2">Use SQL Server Enterprise Manager or SQL Server Management Studio to attach the database located on the LUNs.</td></tr>
<tr><td>If...</td><td>Then...</td></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 9.</td></tr>
<tr><td colspan="3">If you attached the database</td></tr>
<tr><td>4</td><td colspan="2">Start the SnapManager for Microsoft SQL Server application.</td></tr>
<tr><td>5</td><td colspan="2">Click Restore in the Actions pane.

Result: The SnapManager for SQL-Restore dialog box appears.</td></tr>
<tr><td>6</td><td colspan="2">In the Restore to Server box, select the SQL Server whose backup sets you want to use to restore databases to this SQL Server.</td></tr>
<tr><td>7</td><td colspan="2">Click the “...” tab next to the “Point-in-time” option.

Result: The Point-in-time dialog box appears.</td></tr>
</table>

Step	Action
8	Use the “Point-in-time” option to perform an up-to-the-minute restore or a point-in-time restore. ◆ For an up-to-the-minute restore, backup the most recent transactions and select them for restore by selecting the option “Most recent backup selected.” ◆ For a point-in-time restore, select the backup set, a combination of transaction log backups to be restored, or both and select “Committed transactions at the specified time”. For detailed information, follow the steps in “Restoring using the SnapManager Restore option” on page 241. For more information, see the following topics: ◆ For information about identifying the most recent full database backup, see “SnapManager backup set names” on page 155. ◆ For information about identifying the transaction logs, see “SnapInfo subdirectory names” on page 154 and “Transaction log backup” on page 161. The procedure is now complete.

Step	Action
If you did not attach the database	
9	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 backup the currently active transaction log even if the SQL Server 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 backup 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 154 and "Transaction log backup" on page 161. **Attention** Do not use SQL Server Enterprise Manager or SQL Server Management Studio to backup the last active transaction log. Due to file formatting differences between SnapManager Backup 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.
10	Start the SnapManager for Microsoft SQL Server application.
11	Click Restore in the Actions pane. **Result:** The SnapManager for SQL-Restore dialog box appears.

Step	Action
12	In the Restore to Server box, select the SQL Server whose backup sets you want to use to restore databases to this SQL Server.
13	Use the "Point-in-time" option to perform an up-to-the-minute restore or a point-in-time restore. ◆ For an up-to-the-minute restore, backup the most recent transactions and select them for restore by selecting the option "Most recent backup selected." ◆ For a point-in-time restore, select the backup set, a combination of transaction log backups to be restored, or both and select "Committed transactions at the specified time." For detailed information, follow the steps in "Restoring using the SnapManager Restore option" on page 241. For more information, see the following topics: ◆ For information about identifying the most recent full database backup, see "SnapManager backup set names" on page 155. ◆ For information about identifying the transaction logs, see "SnapInfo subdirectory names" on page 154 and "Transaction log backup" on page 161. The procedure is now finished.

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 Server 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	Ensure 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 Actions pane and click Restore Wizard. **Result:** The SnapManager Restore wizard appears and displays the Welcome screen.
8	In the Welcome screen, click Next. **Result:** The SQL Server screen appears.
9	In the SQL Server 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.
10	In the SQL Server Where the Backups were Created screen, do the following: **a.** 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 cleared. **d.** Click Next. **Result:** The Backup Set screen appears and lists the backup sets for the other SQL Server you specified.

Step	Action
11	Use the “Point-in-time” option in the Transaction Logs screen to perform an up-to-the-minute restore or a point-in-time restore. ◆ For an up-to-the-minute restore, backup the most recent transactions and select them for restore by selecting the option “Most recent backup selected.” ◆ For a point-in-time restore, select the backup set, a combination of transaction log backups to be restored, or both and select “Committed transactions at the specified time”. For detailed information, follow the steps in “Restoring using the SnapManager Restore option” on page 241. For more information, see the following topics: ◆ For information about identifying the most recent full database backup, see “SnapManager backup set names” on page 155. ◆ For information about identifying the transaction logs, see “SnapInfo subdirectory names” on page 154 and “Transaction log backup” on page 161. The procedure is now complete.

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: Back up the Last Transaction Log When the Master and the Database Files Are Damaged." This article describes how you can backup the currently active transaction log even if the SQL Server 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 backup 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 154 and "Transaction log backup" on page 161. **Attention** Do not use SQL Server Enterprise Manager or SQL Server Management Studio to backup the last active transaction log. Due to file formatting differences between SnapManager backup 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 Actions pane and select Restore Wizard. **Result:** The SnapManager Restore Wizard appears and displays the Welcome screen.
17	In the Welcome screen, click Next. **Result:** The SQL Server screen appears.
18	In the SQL Server 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.** 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 cleared. **d.** Click Next. **Result:** The Backup Set screen appears and lists the backup sets for the other SQL Server you specified.

Step	Action
20	Use the “Point-in-time” option in the Transaction Logs screen to perform an up-to-the-minute restore or a point-in-time restore. ◆ For an up-to-the-minute restore, backup the most recent transactions and select them for restore by selecting the option “Most recent backup selected”. ◆ For a point-in-time restore, select the backup set, a combination of transaction log backups to be restored, or both and select “Committed transactions at the specified time”. For detailed information, follow the steps in “Restoring using the SnapManager Restore option” on page 241. For more information, see the following topics: ◆ For information about identifying the most recent full database backup, see “SnapManager backup set names” on page 155. ◆ For information about identifying the transaction logs, see “SnapInfo subdirectory names” on page 154 and “Transaction log backup” on page 161. The procedure is now finished.

Restoring system databases from SnapManager backup sets

About this section

The following topics describe the prerequisites and procedures for restoring system databases:

- "Prerequisites for restoring system databases" on page 322
- "Procedures for restoring system databases" on page 322

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 LUNs. For more information, see "How databases are stored on storage system volumes" on page 122.
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 159.

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 must 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 "Restoring Databases Using SnapManager" on page 225.

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 a SQL Server utility:

- For Microsoft SQL Server 2000, see the `Rebuildm.exe` utility to rebuild the system databases. This utility is located in the directory `C:\Program Files\`NetApp `\Microsoft SQL Server\80\Tools\Binn`.

  For more information, see your Microsoft SQL Server 2000 documentation.

For SQL Server 2005 and SQL Server 2008, SQL Server 2008 setup.exe is used. For more information, see SQL Server online help.

- For Microsoft SQL Server 2005 and SQL Server 2008, use the `setup.exe` utility to rebuild the system databases. This utility is located in the directory `C:\Program Files\`NetApp `\Microsoft SQL Server\90\Tools\SetupBootstrap`.

  For more information, see your Microsoft SQL Server documentation.

To rebuild and then restore SQL Server system databases from SnapManager backup sets, complete the following steps.

Step	Action
1	Create a new 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 documentation. **Result:** The system databases are created in the default location.
3	Migrate the system databases from the default location back to the LUN. For more information, see "Understanding control-file based configuration" on page 127.
4	Use SnapManager to restore the system databases from SnapManager backup sets. For more information, see the following topics: ◆ "Understanding SnapManager Restore" on page 228. ◆ "Types of SnapManager restore operations" on page 233. ◆ "Restoring using the SnapManager Restore option" on page 241.

Archiving SnapManager Backups 15

About this chapter

This chapter describes how to create offline archives of Snapshot copies containing SnapManager backup sets. The following topics are covered:

Related topics:

- "Backing Up Databases Using SnapManager" on page 173
- "Performing Disaster Recovery with SnapManager" on page 285

Understanding SnapManager backup set archival

About this section

The following section provide more information about archiving SnapManager backup sets:

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 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 option to check; a backup with a green check mark is verified.
- Create an archive of the most recent backup. For detailed information, see "SnapManager backup set naming conventions" on page 155.
- 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.

Note

If the system is busy, the network is slow, or the load is more on the Data Fabric Manager server or the storage system, there is a time lag between creation of a backup and appearance of the archive in the Restore view.

Choosing the best way to archive

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

About this section

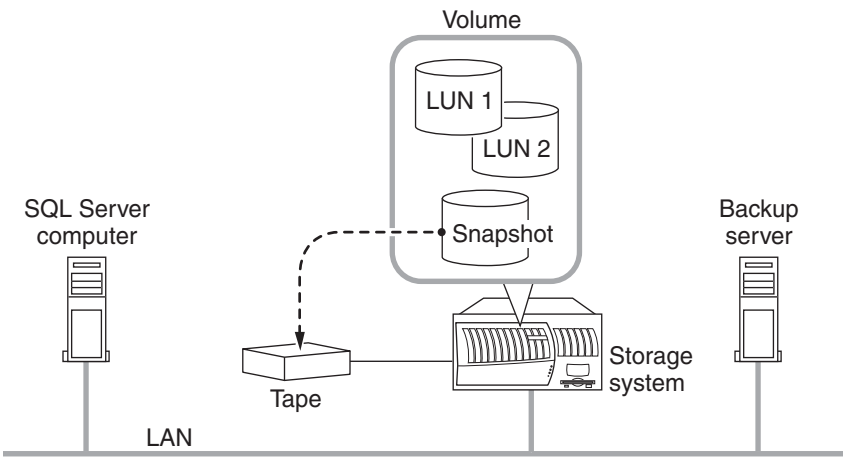
You can use NDMP or the storage system's `dump` command to archive the LUNs containing the SQL Server data and the SnapInfo directory directly from the storage system to the archive medium. NDMP and the `dump` command are the most efficient methods for creating archives of the LUN drive files. See the following topics for more information:

- "About this method" on page 329
- "Guidelines for archiving using NDMP or the dump command" on page 330
- "Procedure for archiving using NDMP or the dump command" on page 331

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 Unmanaged Media mode" to perform the restore operation. This is described in "Recovering SQL Server databases using archives" on page 304.

- 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

About this section

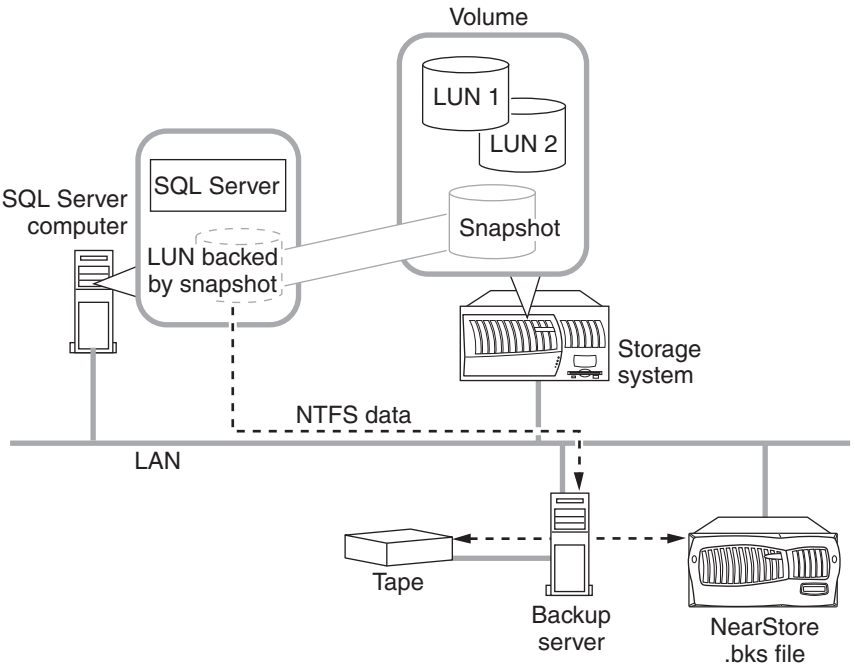
The following topics discuss archiving SnapManager backups using a Windows backup utility:

- "About this method" on page 332
- "Usage example" on page 333

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 222.
- 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 backup 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 backup 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		Sample value passed to the script
Script	**Command line**	
%1	$SqlSnapshot	sqlsnap__winsrvr2__recent
%2	$SnapInfoName	WINSRVR2__recent

Variable		Sample value passed to the script
Script	**Command line**	
%3	$SnapInfoPath	I:\SMSQL_SnapInfo\SQL__WINSRVR2\DB__Northwind

For more information about the Run Command After Operation feature, see "Running a script from a UNC path on a Windows Server 2003 system" on page 360.

Run Command After Operation

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 complete. 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 complete
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 completed 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.

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 Verify option or the Backup Wizard.

Related topics:

- "Archiving SnapManager backups using a Windows backup utility" on page 332
- "Run Command After Operation settings" on page 432

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 a 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 Verify option: To open the Run Command After Operation dialog box while specifying an operation in the Backup and Verify option, 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 settings" on page 432.

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 settings" on page 432.

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 340.
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 337 and "Command arguments are operation specific" on page 338. **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.

Dataset and SnapVault Integration 16

About this chapter

This chapter provides you with information about integrating Protection Manager with SnapManager so that you can create, restore, and manage remote backups. Dataset and SnapVault technologies together form the basis of this integration.

The functionality for dataset and SnapVault integration is available only if you are using Data ONTAP 7.3 RC2 or later along with Protection Manager 3.7 D8 or later.

The following topics are included in this chapter:

Understanding dataset and SnapVault integration

Overview

This topic provides more information about dataset and SnapVault integration with SnapManager. The following topics are covered in this section:

Why dataset and SnapVault integration is required

A dataset is a collection of storage sets with identical data protection requirements on the primary storage system. It is a data management concept introduced in Protection Manager and gives you extensive remote backup and archival capabilities.

The three elements of a dataset are:

- Database
- Protection policy
- Resource pool

The protection policies determine how the data is protected. The resource pool includes the backups and replica of the primary data and its configuration information.

By replicating Snapshot copies to the secondary storage, SnapVault provides you with a centralized disk-based backup solution. It enables you to keep weeks of backup online for faster restore. Through datasets, SnapManager integrates with SnapVault to archive backups to secondary storage.

SnapManager uses Data ONTAP Snapshot technology to create and restore local backups. Dataset and SnapVault integration with SnapManager provides an integrated rapid solution to create and restore remote backup and archives.

SnapManager manages backup on the primary location, but archived backup is managed by Protection Manager.

The following capabilities of Protection Manager make it a good option for integration with SnapManager:

- Automatic setting up of SnapVault relationships and complex replication topologies with resource pools

- Scheduling of remote backups
- Monitoring of data transfer
- Management of remote backup retentions

If Protection Manager is available, and SnapDrive is configured for DataFabric Manager, SnapManager automatically becomes aware of the dataset. If Protection Manager is not available, SnapDrive informs SnapManager of its unavailability. SnapManager continues in the normal working mode, and remote backup is not supported.

Available functionalities

You can do the following with SnapManager integrated with dataset and SnapVault:

- Create and restore remote backup.
- Select policies related to the dataset created by Protection Manager.
- Protect created datasets, by doing the following:
  - Creating remote backup on the SnapVault secondary.
  - Mirroring the local source volume to SnapVault destination volume.
  - Using topologies supported by Protection Manager.
- Delete individual remote backups based on the backup version.
- Display remote backups that are available for restore.
- Perform temporary restore to another location on the secondary storage system using SnapVault remote Snapshot technology.
- Perform remote backup integrity verification.

Limitations

The following are the limitations in integrating SnapManager with dataset and SnapVault:

- No remote backup and archival facility is present if dataset configuration is not available.
- The administrator cannot control the archived backup retention policy through SnapManager. It is controlled by Protection Manager.
- The dataset cannot be used for disaster recovery or business continuance.
- Multiple LUNs residing on the same storage system qtree, and LUNs not residing on a storage qtree, are not supported.
- You need to roll forward archived backed up transaction logs manually.
- System databases are not supported by dataset and SnapVault integration with SnapManager.

Software dependencies

The following are the software dependencies for integrating SnapManager with dataset and SnapVault:

- Protection Manager 3.7 D8 and later
- NetApp Management Console 3.7 and later
- SnapDrive for Windows 6.0.1 and later
- Data ONTAP 7.3 RC2 or later
- SnapVault (for both primary and secondary locations)
- Enable NDMP

You can upgrade SnapManager from an earlier version that did not support datasets to a later version that supports datasets. You can also revert to the older version without any adverse effects on the system.

Prerequisites

The following are the prerequisites for dataset and SnapVault integration with SnapManager:

- Two storage systems should be present.

  One should have the SnapVault primary license, and the other should have the SnapVault secondary license. The primary is the archival source; the secondary is the archival destination. Both should have Data ONTAP 7.3 RC2 or later installed.
- All LUNS must be created on qtrees, and each qtree should contain only a single LUN.
- It is recommended that the Protection Manager software and the NetApp Management Console software should be installed on a dedicated server other than the SQL Server.
- SnapDrive for Windows should be installed.

Integrating dataset and SnapVault to SnapManager

Integrating dataset and SnapVault to SnapManager

Follow this outline of steps to integrate dataset and snapVault to SnapManager.

1. Install Protection Manager on your system.
2. Install SnapDrive for Windows 6.0.1.

   Enable Protection Manager Integration during configuration. For more information, see Protection Manager documentation.
3. Run the Configuration wizard.

   Select the archived backup sets and protection policies.
4. Assign a resource pool to the dataset using Protection Manager.
5. Run SnapManager backup operation.
6. Run SnapManager restore operation.

Configuring datasets

Overview

This section provides information on the following topics:

About dataset configuration

A storage set grouped with its configuration information makes a dataset. Datasets associate the LUNs used by an SQL Server to the related set of protection policies. This enables the administrator to protect the data through remote backup and relate to the corresponding resource pool. One dataset is created for each SQL Server on the server host.

Datasets are created when SnapManager Configuration wizard is run for the first time on a system with Protection Manager installed. If SnapManager is upgraded from an earlier version, rerun the Configuration wizard to setup a dataset. Backups scheduled before the configuration of dataset continue to function without interruption.

The names of the datasets cannot be changed. The following is the example for the naming convention for a dataset:

```
SnapMgr_SQL_server1
```

For SQL Server running on Microsoft Clustered Server, a virtual server is used to name the SnapManager dataset.

About protection policies

The dataset policies control the protection of data in dataset. A policy decides the following characteristics:

- Data replication topology
  - SnapVault topology (also called Backup topology)
- Backup retention type

- Primary (Determined using SnapManager, Protection Manager remains unaffected)
- Secondary (Determined using Protection Manager)

- Replication lag and throttle

After a dataset policy is set up, it cannot be changed to another policy from SnapManager. If it is changed using Protection Manager, it is automatically picked up by SnapManager.

Each dataset has a policy assigned to it. But a single policy may be applied to many datasets. Hence modifying a policy might affect all the associated datasets.

You can create a new policy by modifying an existing policy using Protection Manager. For more information, see Protection Manager documentation.

Note

"Remote backups only" policy is the policy that SnapManager currently supports.

Remote backup retention policies

Remote backup retention policies control the backups created at the remote site. The remote backup retention policies are controlled by SnapDrive and Protection Manager, not SnapManager.

Creating a dataset using SnapManager

You can create a dataset to manage protection for data that shares the same protection requirements. For one SQL server, there can be only one dataset. Create this dataset when you run SnapManager Configuration wizard with Protection Manager for the first time.

Before you begin, ensure that you are assigned an administrator role that enables you to create a dataset. Also ensure that the primary databases are configured properly before the archival process is carried out, or it will fail.

Editing a dataset using Protection Manager

After the dataset is created using SnapManager, check the Protection status and the Conformance status of the dataset using Protection Manager. Next, you need to add the resources at the secondary storage system manually using Protection Manager. To add the resources, follow these steps.

Step	Action
1	Open the NetApp Management Console.
2	In the left pane, click Data. **Result:** The dataset pane appears.
3	In the Data Set pane, click the Edit icon. **Result:** The Edit Data Set window appears.
4	In the left pane, click Physical Resources or Provisioning / Resource Pools.
5	From the list of available resources, add the resources in the Data Set.
6	Click Next. **Result:** The Completing the Data Set Node Wizard window appears.
7	Click Finish.

SnapVault relationships

After the dataset is created, policies are determined, and secondary resource pools are added to the dataset, Protection Manager creates SnapVault relationships for archiving. A remote backup restore is not possible if the SnapVault relationship is changed or modified.

If you already have an existing SnapVault relationship, Protection Manager cannot use the existing SnapVault relationship for the dataset automatically. Import the existing SnapVault relationship using the “Import Relationship Wizard” in Protection Manager. For more information, see Protection Manager documentation.

If you do not import the SnapVault relationship, a new one is created. For more information, see Protection Manager documentation.

If you have a SnapVault relationship for the LUN that is used by database, deleting the SnapVault base line Snapshot copy will result in a SnapDrive for Windows error.

Dataset member information

The dataset member information is a list of drive letters and mountpoints related to SnapManager. It is stored and tracked by Protection Manager, and its information is retained even after SnapDrive is uninstalled. The member information is retained on all cluster nodes.

Protecting local backups

Overview

By creating remote backups, SnapManager uses datasets to protect the local backups that were created at the primary storage system. The following conditions should be met before SnapManager starts creating backups:

- A dataset is created.
- The "Archive local backup using SnapVault" option is enabled.
- The dataset has the protection status as "Protected" and conformance status as "Conformant."

  If the configuration contains non-SQL LUNs, the qtrees containing the non-SQL database are not updated during archiving. This changes the dataset protection status to "Lag Warning" or "Lag Error". For more information, see Protection Manager documentation.

The following topics are discussed in this section:

- "Creation of remote backup" on page 353
- "Remote backup retention" on page 354

Creation of remote backup

The process of remote backup starts after local backups are created. SnapManager conveys the following information to SnapDrive before actuating the remote backup process:

- The version number of the backup

  The version number acts as the time stamp for the backup and is used by SnapManager to retrieve detailed information about the backup during restore.
- The backup management group

  Two types of management groups are available:

  - Local management group

    The local management groups can be standard, daily and weekly
  - Remote management group

    The remote management groups can be hourly, daily, and weekly, monthly, all, and unlimited. The default management group is daily.

    If you select the hourly management group for remote backup, SnapManager shows a message conveying that hourly archived backups are deleted when Protection Manager restarts.

- A list of LUNs with their corresponding Snapshot names

You can defer remote backup for some time after the local backup is created.

In the Backup wizard, if dataset is configured and the archival process is initiated, the generic backup naming convention is automatically changed to the unique backup naming convention. If you choose to keep the naming convention as generic, no archives are created.

Remote backup retention

Remote backup retention capability refers to the number of backups that can be retained at the secondary storage system. You can determine the number by using the backup management groups. Remote backup retention is controlled by Protection Manager. When SnapManager deletes a backup, it deletes the metadata only after confirming with Protection Manager that the archive backup has also been deleted. The SnapInfo directory that retains the backup metadata in the live file system is not deleted, even if the local backup has been deleted.

When Protection Manager applies the remote backup retention policy to the dataset, it deletes the older version of backup. New backups are continuously created. If the number of backups or days exceed the management group setup, the policy deletes the last backup at the secondary storage system.

Retrieving and restoring remote backups

Overview

This section provides information on the following topics:

Retrieving remote backups

To retrieve a remote backup, SnapManager uses the name of the dataset and the SnapInfo directory to create a list of databases that need to be restored.

Restoring remote backups

The process of restoring a remote backup is almost the same as restoring a local backup, except that the remote backup needs to be restored from the archived backup and the backed up transaction logs have to be applied manually.

To restore an archived database from the Restore wizard, follow these steps.

Step	Action
1	Select the SQL Server from the Scope pane.
2	Select Restore Wizard from the Actions pane. **Result:** The Restore wizard opens.
3	Click Next. **Result:** The "Which SQL Server Created the Backups" window appears.
4	In the "Which SQL Server Created the Backups" window, select the option "Restore from archive."
5	Continue with the instructions provided in the Restore wizard.
6	Click Finish to restore the database from the archived backups.

Deferred database integrity verification

Deferred integrity verification can be performed on the local and the remote backup. Deferred integrity verification can be carried out in two ways:

- Deferred integrity verification at the local location

  SnapManager archives the local backup first, then verifies the backup on the SnapVault secondary storage system.

  Both “Archive backup to secondary storage” and “Verify on archived backup on secondary storage” should be enabled.

  Both local and remote backup management groups are used. The local backups of the local backup management group are archived using the remote backup management group.

- Deferred integrity verification at the archived location (secondary storage system)

  SnapManager runs verification on the backups already archived on the SnapVault secondary storage.

  Only “Verify on archived backup on secondary storage” should be enabled, and only the backups of the remote backup management group are verified.

Deferred integrity verification runs on the management group that you selected previously. The remote backup management group can always be changed after remote backup is created.

You can perform deferred verification on the SnapVault secondary storage system from both the local application server and the remote verification server.

Types of restore operations supported with dataset and SnapVault integration

The following table describes the restore operations supported with dataset and SnapVault integration with SnapManager.

Operation	Backup type	Restore from archive
Restore system databases	Not supported	Not supported
Restore user databases	Full backup	Automatic
	Transaction log backup	Manual

Operation	Backup type	Restore from archive
Backup verification during restore	Full backup	Automatic
	Transaction log backup	Not supported
Restore when the system databases and the user databases share one LUN	Full backup and transaction log backup	Not supported
Clone user databases	Full backup	Automatic
	Transaction log backup	Manual

Deleting archived backups

Deleting archived backups

The process of deleting archived backups is the same as deleting local backups. Be sure to read the following points before deleting archived backups:

- When a local backup is deleted, SnapManager does not delete the backup metadata and SnapInfo file in the SnapInfo directory but deletes the transaction logs in the SnapInfo directory.
- If SnapManager is unable to find both the local and archived backups, it deletes the backup metadata and the SnapInfo directory associated with the backup.

Note

Make sure that Protection Manager is always available, otherwise SnapManager deletes the backup metadata for the archived backups.

Tools for Managing Backup and Verification A

About this chapter

This chapter 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:

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:

- "Understanding SnapManager Backup Sets" on page 149
- "Backing Up Databases Using SnapManager" on page 173
- "Run Command After Operation settings" on page 432

Running a script from a UNC path on a Windows Server 2003 system

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 Uniform Naming Convention (UNC) 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 in workgroup mode” on page 35.
2	Launch Internet Explorer.
3	In the menu bar, select Tools > Internet Options. **Result:** The Internet Options dialog box appears.

Step	Action
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.
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

About this section

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 362
- "Using the Schedule Job dialog box" on page 362

Related topics:

- "Creating a full database backup using SnapManager" on page 176
- "Creating a transaction log backup using SnapManager" on page 190
- "Performing database verification using SnapManager" on page 202

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 418.

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 Backup and Verify 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 “Run Command After Operation settings” on page 432.

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.</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 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 a 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>
</table>

SnapManager Command-Line Reference B

About this chapter

This appendix provides details about the SnapManager operations you can execute using the new SnapManager command-line functionality. This new function allows you to create scripts to run SnapManager functionality without using the SnapManager graphical user interface (GUI).

The following topics are covered in this section:

Guidelines for using the command-line utility

Location of the SnapManager PowerShell

To launch SnapManager PowerShell, go to Start > Programs > NetApp > SnapManager PowerShell.

Common parameters used

The following are the ubiquitous (common) parameters in PowerShell:

Debug (-db): This parameter displays the debug information for the cmdlet used.

ErrorAction Action Preference (-ea): Scripting blocks use this parameter. The following are the examples that explain the usage of this parameter.

- `SilentlyContinue:` Continue without printing.
- `Continue:` Print and then continue (This is the default setting.)
- `Stop:` Halt the command or script.
- `Inquire`: Ask the user what to do.

ErrorVariable (-ev): This parameter displays the error data in the specified variable.

OutVariable (-ov): This parameter displays the output data string.

OutBuffer (-ob): This parameter displays the output buffer.

Whatif: This parameter gives you a preview of an operation.

Confirm: This parameter prompts you for confirmation before the actual deletion operation starts.

Verbose (-vb): This parameter displays the report content for backup, restore, configuration, and verification options

Tips for using the command-line interface

Observe the following guidelines when using the SnapManager command-line functionality:

- All parameters and options are case-insensitive. For example, if you use the option `-Daily,` it achieves the same results as you get if you use `daily`.

- Some of the options must be invoked in a particular order. For best results, use the order specified in the syntax 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`.
- Press Ctrl-D to cancel a running operation. Closing the PowerShell window does not cancel the running operation.

If the execution policies in your system are restricted, you might be unable to load the PowerShell snap-in. To check and reset the execution policies on your system, follow these steps:

Step	Action
1	Enter the command `get-executionpolicy` in PowerShell.

2	If the policy displayed is “Allsigned” or “Restricted”, enter any of the following commands:
	`set-executionpolicy unrestricted`
	or
	`set-executionpolicy remotesigned`

new-backup

Name

new-backup

Synopsis

This cmdlet enables you to back up the SQL server databases in SnapManager PowerShell command-line interface.

Detailed description

This cmdlet enables you to begin the backup-only and backup-with-verify operations. SnapManager provides a separate cmdlet for verification. You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
new-backup [-Server <ServerName>]
[-Database <<SQL server instance>,<Number of
databases>,<db1>,<db2>,db3>..>]
[-logbkup][-Username <username>] [-Password <password>]
[-Verify] [VerifyServerInstance <SQLServerInstance>]
[-VerSvrLogin <LoginName>] [-VerSvrPassword <Password>]
[-RetainBackups <number of backups>] [-VerDestVolume]
[-VerifyOnDestVolumes <comma-separated list of source storage
system, source volume, destination storage system, destination
volume>]
[-MountPointDir <directory pathname>] [-UseMountPoint]
[-UpdateMirror]
[-Managementgroup <system string [Daily, Weekly, Standard]>]
[-RetainSnapofSnapInfo [<Number of SnapInfo Snapshots to keep>]]
[-LogBkupOnly][-BkUpSIF]]
[-TruncateLogs] [-AttachDB]
[-NoRetainUTM] [-RunDBCCAfter] [-RunDBCCBefore] [-Command]
[-Runcommand <script file pathname>]
```

```
[-CmdArguments <List of Command Arguments>]
[-commandserver <ServerName>] [-GenericNaming] [-ArchiveBackup]
[-VerifyArchiveBackup]
[-ArchivedBackupRetention <Daily/hourly/weekly/monthly/unlimited>]
[-RetainSnapofSnapInfoDays <Number of days>]
[-RetainBackupDays <Integer>][<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

```
-svr win-225-161
```

```
-svr sql1
```

-Database (String Parameter): Use this parameter to specify the original database that you want to restore. You can also specify multiple database names only if the databases share a single LUN or multiple LUNs together. In this case, list the databases followed by -Database in following format:

```
-Database "DatabaseName1"," DatabaseName2"
```

This is a required parameter. If you do not specify the database parameter explicitly, the cmdlet backs up all the databases from all the SQL server instances in the host. Also, if a non-NetApp storage exists on your system, the cmdlet fails if you do not specify databases explicitly using this parameter.

-Logbkup (Switch Parameter): If you are performing a backup of the full database, use this option to backup the transaction logs after the full database backup completes.

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified.

-Verify (Switch Parameter): Use this parameter to verify the backed up databases and logs. It depends on the type of backup chosen. If the type is full database backup it verifies the full database and if the type is transaction log backup it verifies transaction logs.

-VerifyServerInstance (String Parameter): This parameter specifies the separate SQL server that is used to run the Database Consistency check utility. If you have not specified the -verify parameter, SnapManager ignores this parameter.

The following example illustrates the usage:

```
-verInst win-225-161
```

Here the SQL server instance is the local or remote SQL server instance to verify on. SnapManager takes the configured SQL server instance that is used for verify in client configuration (registry) as the default SQL server instance.

-VerSvrLogin (String Parameter): This parameter specifies that SQL Server authentication is used. If you do not specify the login name, SnapManager uses Windows NT Authentication.

-VerSvrPassword (String Parameter): SnapManager uses this parameter to verify the user credentials before creating a backup job. If you do not specify the VerSvrLogin parameter, SnapManager ignores this parameter.

-RetainBackups (Integer): This parameter specifies the number of backups that need to be retained after the delete phase.

-VerDestVolume (Switch Parameter): This parameter enables the verification of the SnapMirror destination volume. It carries the “false” value by default.

-VerifyOnDestVolumes (String Parameter): Specify this parameter to override the default SnapMirror relationships. Enter a comma-separated list of the source storage system, the source volume, the destination storage system, and the destination volume.

-MountPointDir (Integer): Use this parameter to specify the mount point directory on which a backup set is mounted during database verification. This parameter should be used along with the parameter -UseMountPoint.

-UseMountPoint (Switch Parameter): This parameter is a switch which specifies that the Snapshot copy must be mounted to an NTFS directory. During a SnapManager verification operation, Snapshot copies are mounted to the default NTFS directory for database verification. The option is effective when there are no available drive letters to mount the Snapshot copies. It overrides preconfigured SnapManager verification settings.

-UpdateMirror (Switch Parameter): Use this option to update the SnapMirror destination after the backup or verification operations are complete, if you are using backups that reside on volumes configured as SnapMirror sources.

-ManagementGroup (String Parameter): This parameter denotes the backup or verify operation that SnapManager performs on daily, or weekly, or standard basis. The default management group is standard.

-RetainSnapofSnapinfo (Integer): Use this option if you want to delete the oldest Snapshot copies in the SnapInfo directory, specified that the backup type is a transaction log backup only. It has an integer value. The following example illustrates the usage of this parameter:

```
-rtsifsnap Number of SnapInfo Snapshots to keep
```

Note

This option is valid only if you specify the parameter – BkupSIF.

-LogBkupOnly (Switch Parameter): Use this option to back up your SQL server transaction log files.

-BkupSIF (Switch Parameter): Use this option to create a Snapshot copy of the SnapInfo directory after the backup of the transaction log completes. The backup type should be a transaction log backup only.

-TruncateLogs (Switch Parameter): This parameter is a switch, which if specified allows you to truncate the backed up transaction logs. Use this option to conserve space on the LUN containing the backed up transaction logs. SnapManager uses the value "false" by default.

-AttachDB (Switch Parameter): If the operation includes a database or transaction log verification, use this option when you want to specify that the databases are to be attached after the verification.

-NoRetainUTM (Switch Parameter): Use this option if you do not want to retain up-to-the-minute restore ability for older backups in other management groups.

-RunDBCCAfter (Switch Parameter): Use this option when you want to verify a live database after database backups are performed.

-RunDBCCBefore (Switch Parameter): Use this option when you want to verify a live database before the database backups are performed.

-**Command (Switch Parameter):** This is a switch parameter that indicates run command after operation.

-Runcommand (String Parameter): This parameter runs the specified command after the SnapManager backup or verification operation is complete. It defines the complete path for the command to be run after the backup or verify operation is complete.

Note

You need to specify this command explicitly, the pre configured command does not run after the backup or verification operation.

-CmdArguments (String Parameter): This option contains the string of SnapManager operation-specific information to be passed to your program or script. It is considered only if Command and Run Command are specified.

-Commandserver (String Parameter): This option specifies the server on which the command is to be run after the backup or verify operation is complete. It is considered only if Command and Run Command are specified.

-GenericNaming (Switch Parameter): This parameter sets the naming convention for new backups as generic.

-ArchiveBackup (Switch Parameter): Use this parameter to archive database to a secondary storage system.

-VerifyArchiveBackup (Switch Parameter): Use this parameter to verify database archived at the secondary storage system.

-ArchivedBackupRetention (String Parameter): Use this parameter to specify whether you want to retain backups at the archived location on a daily, hourly, weekly, monthly or unlimited basis.

-RetainSnapofSnapInfoDays (Integer): Use this parameter to delete SnapInfo Snapshot copies older than the specified number of days. This parameter is mutually exclusive with the parameter RetainSnapofSnapinfo and they cannot be specified together in the same cmdlet.

-RetainBackupDays (Integer): Use this parameter to specify the number of days you want to retain the backups for. SnapManager deletes backups older than the specified number of days. The parameters RetainBackups and RetainBackupDays are mutually exclusive and cannot be specified together.

Short names

The following table lists the short names of the parameters.

Parameter name	Short name
Server	-svr
Database	-d
Logbkup	-lb
Username	-usr
Password	-pwd
UseMountPoint	-mp
Verify	-ver
VerDestVolume	-verdest
VerOnDestVolumes	-vermirror
VerifyServerInstance	-verInst
UpdateMirror	-updmir
AttachDB	-attdb
VerSvrLogin	-verlogin
VerSvrPassword	-verpwd
RetainBackups	-rtbackups
Managementgroup	-mgmt
LogBkupOnly	-lgbkonly
BkupSIF	-bksif
RetainSnapofSnapinfo	-rtsifbkup
TruncateLogs	-trunclog
NoRetainUTM	-noutm
RunDBCCAfter	-dbccaf
RunDBCCBefore	-dbccbf
Command	-cmd

Parameter name	Short name
Runcommand	-runcmd
CmdArguments	-cmdargs
commandserver	-cmdsvr
GenericNaming	-generic
RetainSnapofSnapInfoDays	-rtsifsnapdays
ArchiveBackup	-arch
VerifyArchiveBackup	-verarch
ArchivedBackupRetention	-archret
RetainBackupDays	-rtdays

Examples

Example 1: `new-backup -Server 'DBServer1' -Verify -VerifyServerInstance 'Snapmgr-50'`

This command creates a backup of all databases on the host DBServer1 and verifies the backups using the remote server Snapmgr-50.

Example 2: `new-backup -svr 'VM-VS-1' -d 'VM-VS-1', '4', 'ds_test1', 'ds_test2', 'ds_test6', 'ds_test7' -ver -verInst 'ZEUS-VM1\VERSERVER' -rtbackups 7 -lb -bksif -rtsifsnap 8 -trlog -noutm -mgmt standard -ArchiveBackup -VerifyArchiveBackup -ArchivedBackupRetention daily`

This example illustrates the creation of a new backup with verification of local backups and archive backups.

Example 3: `new-backup -svr 'VM-VS-1' -d 'VM-VS-1', '2', 'model', 'sm_test' -ver -verInst 'ZEUS-VM1\VERSERVER' -rtbackups 7 -lb -bksif -rtsifsnap 8 -trlog -noutm -gen -mgmt standard`

This example creates a new backup with the generic naming convention.

Example 4: `new-backup -svr 'VM-VS-1' -d 'VM-VS-1', '2', 'model', 'sm_test' -ver -verInst 'ZEUS-VM1\VERSERVER' -rtbackups 7 -lb -bksif -rtsifsnap 8 -trlog -noutm -mgmt standard`

This example creates a new backup with the unique naming convention.

verify-backup

Name

verify-backup

Synopsis

This cmdlet enables you to verify the SQL server databases in SnapManager PowerShell command-line interface.

Detailed description

This cmdlet enables you to perform verification operations. You can mount the Snapshot copies, manage SnapMirror relationships and destinations, assign management groups for verification and so on.

You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
verify-backup [-Server <ServerName>]
[-Database <<SQL server instance>,<Number of
databases>,<db1>,<db2>,db3>..>]
[-Username <UserName>] [-Password <Password>][-UseMountPoint]
[-MountPointDir <directory pathname>][-VerDestVolume]
[VerifyServerInstance <SQLServerInstance>]
[-VerifyOnDestVolumes <Comma separated list of source storage
system, source volume, destination storage system,destination
volume>][-UpdateMirror][-AttachDB]
[-VerifyServerInstance <SQL Server Instance>]
[-VerSvrLogin <LoginName>][-VerSvrPassword <Password>]
[-BackupNo <Number of recent unverified backups to verify>]
[-Managementgroup <system string [Daily, weekly, standard]>]
[-Command] [-Runcommand<Command>]
[-CmdArguments <List of Command Arguments>]
```

```
[-commandserver <ServerName>] [-ArchiveBackup]
[-VerifyArchiveBackup]
[-ArchivedBackupRetention <Daily/hourly/weekly/monthly/unlimited>]
[<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

```
-svr win-225-161
```

```
-svr sql1
```

-Database (String Parameter): This parameter specifies the list of databases that are separated by commas. If you do not specify the database parameter, the cmdlet verifies all the databases from all the SQL server instances in the host.

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified.

-UseMountPoint (Switch Parameter): This parameter specifies that the Snapshot copy must be mounted to an NTFS directory. During a SnapManager verification operation, Snapshot copies are mounted to the default NTFS directory for database verification. The option is effective when there are no available drive letters to mount the Snapshot copies. It overrides pre-configured SnapManager verification settings.

-MountPointDir (String Parameter): Use this parameter to specify the mount point directory on which a backup set will be mounted during database verification.

-VerDestVolume (Switch Parameter): Use this parameter to verify the database on the SnapMirror destination volume. This parameter is set to "false" by default.

-VerifyOnDestVolumes (String Parameter): Specify this parameter to override the default SnapMirror relationships. Enter the source and destination storage systems and volumes as a comma-separated list.

-UpdateMirror (Switch Parameter): Use this option to update the SnapMirror destination after the backup or verification operations are complete, if you are using backups that reside on volumes configured as SnapMirror sources.

-AttachDB (Switch Parameter): If the operation includes a database or transaction log verification, use this option when you want to specify that the databases are to be attached after the verification.

-VerSvrLogin (Switch Parameter): This parameter specifies that SQL Server authentication is used. If not specified, the default Windows NT Authentication mechanism is used.

-VerSvrPassword (String Parameter): If the parameter -VerSvrLogin is not specified this parameter will be ignored.

-BackupNo (Integer): This option specifies the number of most recent unverified backups to verify. It is an integer with a default value of 1.

-ManagementGroup (String Parameter): This parameter denotes the verify operation that is performed on daily, or weekly, or standard basis. The default management group is standard.

-**Command (Switch Parameter):** This is a switch parameter that indicates run command after operation.

-RunCommand (String Parameter): This parameter runs the specified command after the SnapManager backup or verification operation is complete. It defines the complete path for the command to be run after the backup or verify operation is complete. SnapManager ignores the -RunCommand parameter if you do not specify the parameter -Command and SnapManager runs the preconfigured command.

-CmdArguments (String Parameter): This parameter contains a string of SnapManager operation-specific information to be passed to your program or script. It is considered only if the parameters -Command and -RunCommand are specified.

-CommandServer (String Parameter): This parameter specifies the server on which the command is to be run after the backup or verify operation is complete. It is considered only if the parameters -Command and -RunCommand are specified.

-ArchiveBackup (Switch Parameter): Use this parameter to archive database to a secondary storage system.

-VerifyArchiveBackup (Switch Parameter): Use this parameter to verify database archived at the secondary storage system.

-ArchivedBackupRetention (String Parameter): Use this parameter to specify whether you want to retain backups at the archived location on a daily, hourly, weekly, monthly, or unlimited basis.

-VerifyServerInstance (String Parameter): This parameter specifies the separate SQL server that is used to run the Database Consistency check utility. If you have not specified the -verify parameter, SnapManager ignores this parameter.

The following example illustrates the usage:

```
-verInst win-225-161
```

Here the SQL server instance is the local or remote SQL server instance to verify on. SnapManager takes the configured SQL server instance that is used for verify in client configuration (registry) as the default SQL server instance.

Short names

The following table lists the short names of the parameters.

Parameter name	Short name
Server	-svr
Database	-d
Username	-usr
Password	-pwd
UseMountPoint	-mp
MountPointDir	-mpdir
VerDestVolume	-verdest
VerOnDestVolumes	-vermirror
UpdateMirror	-updmir
AttachDB	-attdb
VerSvrLogin	-verlogin
VerSvrPassword	-verpwd

Parameter name	Short name
BackupNo	-bkno
Managementgroup	-mgmt
Command	-cmd
Runcommand	-runcmd
CmdArguments	-cmdargs
Commandserver	-cmdsvr
ArchiveBackup	-arch
VerifyArchiveBackup	-verarch
ArchivedBackupRetention	-archret
VerifyServerInstance	-verInst

Examples

Example 1: `verify-backup -svr 'VM-VS-1' -d 'VM-VS-1', '2', 'ds_test6', 'ds_test7' -verInst 'ZEUS-VM1\VERSERVER' -bkno 1 -mgmt standard -ArchiveBackup -VerifyArchiveBackup -ArchivedBackupRetention Daily`

This command initiates deferred verification for the specified database at the specified server, with one unverified most recent backup. The management groups is standard.

restore-backup

Name

restore-backup

Synopsis

This command enables you to restore backed up databases using SnapManager for SQL PowerShell command-line interface.

Detailed description

This cmdlet enables you to restore a database to an alternate location, or to an archive. It also gives point-in-time restore, verification, force restore and many other options.

You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
restore-backup [-Server <ServerName>] [-Username <username>]
[-Password <password>]
[-ServerInstance <comma-separated list of server instances to
restore>]
[-RestoreFromUnmanagedMedia]
[-SnapInfoDirectory <Directory pathname>]
[-Database <comma-separated list of original database names to
restore>]
[-Backup <Backup set name>][-PointInTime <comma separated list of
date-times>]
[-VerDestVolume]
[-VerifyOnDestVolumes <Source storage system, source volume,
destination storage system, destination volume>]
[-VerifyServerInstance <SQL Server Instance>]
[-VerSvrLogin <UserName>][-VerSvrPassword <Password>]
[-VerifyDisable][-ForceRestore]
[-RecoverDatabase]
```

```
[-TransLogsToApply <Comma separated list of transaction logs to
restore>]
[-RestoreLastBackup <Integer>]
[-RestoreArchivedBackup] [-NoAccessToRemoteBackup]
[-ProxyServer <Proxy server name>] [-ArchiveBackup]
[<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

```
-svr win-225-161
```

```
-svr sql1
```

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified.

-ServerInstance (String Array): This parameter specifies the SQL server instance where the database is backed up originally. SnapManager takes the local computer name as the default server instance.

You can specify multiple server instance names here as a comma-separated list. If multiple databases reside on the same LUN but are owned by different SQL server instances when you backed them up originally, use the following format:

```
-Inst "SQLServerInstance1","SQLServerInstance2"
```

The first database specified in the -Database parameter refers the first server instance in the -ServerInstance parameter, the second database in the -Database parameter refers to the second server instance in the -ServerInstance parameter and so on.

-RestoreFromUnmanagedMedia (Switch Parameter): Use this parameter if you are restoring databases from archived SnapManager backup sets.

-SnapInfoDirectory (String Parameter): Use this parameter to specify the SnapInfo directory path of the archived backup set. Use the parameter only along with the -RestoreFromUnmanagedMedia parameter.

-Database (String Parameter): Use this parameter to specify the original database that you want to restore. You can specify multiple database names using this option if the databases share a single LUN or multiple LUNs together. Use the following format:

```
-Database "DatabaseName1"," DatabaseName2"
```

Note

All the databases selected should be present in the selected Snapshot copy.

-Backup (String Parameter): Use this option to specify the name of the backup set. The following example illustrates the usage:

```
-bkup sqlsnap__SYMNASQLDEV170_04-11-2007_15.22.27
```

-PointInTime (String Parameter): Use this switch to restore databases until a specific point in time. The format for the point-in-time string is yyyy-mm-ddThh:mm:ss, with time specified in a 24-hour format.

In case of multiple databases you should specify the point-in-time values for every database separated by a comma. The number of values after the parameter name should equal the number of databases selected. The first value will be applied to the first database specified after the –Database parameter, the second value to the second database, and so on. The following example illustrates the usage:

```
-pit 2008-10-22T11:50:00, 2008-11-25T22:50:00
```

Note

The parameter correspondence is one-to-one, that is, the first point-in-time parameter value specified after the parameter -pit is applied to the first database specified in the parameter -Database and the second point-in-time parameter value to second database and so on.
The values should conform to the required PointInTime regular expression.

-VerDestVolume (Switch Parameter): Use this parameter to verify the database on the SnapMirror destination volume. SnapManager sets it to "false" by default.

-VerifyOnDestVolumes (String Parameter): Specify this parameter in order to override the default SnapMirror relationships. Enter the source and destination storage systems and volumes as a comma-separated list. SnapManager sets it to "false" by default.

-VerifyServerInstance (String Parameter): This parameter specifies the separate SQL server that is used to run the Database Consistency Check (DBCC) utility. If you have not specified the -verify parameter, SnapManager ignores this parameter.

The following example illustrates the usage:

```
-verInst win-225-161
```

Here the SQL server instance is the local or remote SQL server instance to verify on. SnapManager takes the configured SQL server instance that is used for verify in client configuration (registry) as the default SQL server instance.

-VerSvrLogin (String Parameter): This parameter specifies that SQL Server authentication is used. If not specified, the default Windows NT Authentication mechanism is used.

-VerSvrPassword (String Parameter): SnapManager ignores this parameter if the parameter -VerSvrLogin is not specified.

-VerifyDisable (Switch Parameter): This parameter overrides verification and can disable verification even if the database was not verified after backup.

-ForceRestore (Boolean Parameter): Use this parameter to force the restore of a database based on its state. SnapManager sets its value to "true" by default.

-RecoverDatabase (Boolean Parameter): Use this parameter if you want to recover the database after the restore operation. SnapManager sets its value to "true" by default.

-TransLogsToApply (String Parameter): This parameter specifies the list of transactions logs that need to be applied. SnapManager applies all transaction logs of the databases specified in the -Database parameter by default. You can specify the number of transaction logs to be applied for every database mentioned in the -Database parameter. The list of number of transaction logs that have to applied has to be listed in the same sequence as the databases listed in the -Database parameter. For example,

```
restore-backup -svr MACHINE1\INST1 -database db1,db2
-TransLogsToApply 3,7
```

-RestoreArchivedBackup (Switch Parameter): Use this parameter to restore database from an archived backup.

-ProxyServer (String Parameter): This parameter defines the name of the proxy server. Use it along with the parameter NoAccessToRemoteBackup.

-NoAccessToRemoteBackup (Switch Parameter): This parameter specifies that there is no direct access to the secondary storage system. SnapManager uses the proxy server to access the secondary storage system.

-RestoreLastBackup (Integer): Use this parameter to restore backups without specifying the name. If you try to use the Backup and RestoreLastBackup parameters together, SnapManager ignores the RestoreLastBackup parameter and uses the Backup parameter during restore operation. A typical usage example of the restorelastbackup parameter is as follows:

```
restore-backup -restorelastbackup 1
```

Note

If the value for RestoreLastBackup parameter is 0, SnapManager restores the latest backup. If the value is 1, SnapManager restores second-to-latest backup and so on.

Short names

The following table lists the short names of the parameters.

Parameter name	Short name
Server	-svr
Database	-d
Username	-usr
Password	-pwd
ServerInstance	-inst
RestoreFromUnmanagedMedia	-rstumm
SnapInfoDirectory	-snapinfo
Backup	-bkup
PointInTime	-pit
Verify	-ver

Parameter name	Short name
VerDestVolume	-verdest
VerOnDestVolumes	-vermirror
VerSvrLogin	-verlogin
VerSvrPassword	-verpwd
ForceRestore	-force
VerifyDisable	-verdis
RecoverDatabase	-recoverdb
TransLogsToApply	-translogs
RestoreLastBackup	-lastBkup
VerifyServerInstance	-verInst
RestoreArchivedBackup	-rstarchbkup
NoAccessToRemoteBackup	-noaccessarchivebkup
ProxyServer	-pxy

Examples

Example 1: `restore-backup -Server sql1 -Database "Db1"`

This command restores the backup of database Db1 on SQL server sql1.

Example 2: `Restore-backup -svr 'VM-VS-1' -inst vm-vs-1 -d 'ds_test7' -backup sqlsnap__VM-VS-1_07-18-2008_03.19.14__Daily`

This example restores the specified backup on the given server instance.

get-backup

Name

get-backup

Synopsis

This cmdlet enables you to list the backup sets of a particular database using the SnapManager SQL Server PowerShell command-line interface.

Detailed description

This cmdlet enables you to list the backup sets of a particular database by specifying an SQL server, an SQL server instance, or a database set. You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
get-backup [-Server <ServerName>] [-Database <<SQL server
instance>,<Number of databases>,<db1>,<db2>,db3>..>
[-Username <UserName>] [-Password <Password>]
[-ServerInstance <SQL Server Instance Name>] [<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

`-svr win-225-161`

`-svr sql1`

For virtual server instances, specify the virtual server name. For example,

```
get-backup -server <virtual_server> -ServerInstance
<virtual_instance> -d aa1
```

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified.

-ServerInstance (String Parameter): This parameter specifies the SQL server instance where the database is backed up originally. SnapManager takes the local computer name as the default server instance.

-Database (String Parameter): This parameter specifies the list of databases that are separated by commas. If you do not specify the database parameter, the cmdlet backs up all the databases from all the SQL server instances in the host.

Short names

The following table lists the short names of the parameters.

Parameter Name	Short name
Server	-svr
Username	-usr
Password	-pwd
ServerInstance	-inst
Database	-d

Examples

Example 1: `get-backup -svr 'VM-VS-1' -inst vm-vs-1 -d 'ds_test7'`

This example retrieves the backed up database on a server instance of the specified server.

delete-backup

Name

delete-backup

Synopsis

This cmdlet enables you to delete the SnapManager backup sets using the SnapManager SQL Server PowerShell command-line interface.

Detailed description

This cmdlet enables you to delete a database depending on the input criteria specified in the command-line interface. It deletes the specified backup set if it contains the specified database name.

You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
delete-backup [-Server <ServerName>] [-Database <<SQL server
instance>,<Number of databases>,<db1>,<db2>,db3>..>
[-Username <UserName>] [-Password <Password>]
[-ServerInstance <SQL Server Instance Name>]
[-ArchiveBackup <Archive Backup set name>]
[-Backup <Backup set Name>] [<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name. In case of a clustered configuration, the virtual server name is the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

```
-svr win-225-161

-svr sql1
```

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified.

-ServerInstance (String Parameter): This parameter specifies the SQL server instance where the database is backed up originally. SnapManager takes the local computer name as the default server instance.

-Database (String Parameter): This is a mandatory parameter that specifies the list of databases that are separated by commas. If you do not specify the database parameter, the cmdlet backs up all the databases from all the SQL server instances in the host.

-ArchiveBackup (String Parameter): Use this parameter to specify the archived backup set that needs to be deleted.

Note
This parameter is mandatory if you delete archived backup sets.

-Backup (String Parameter): Use this parameter to specify the backup set that needs to be deleted. It is a mandatory parameter.

Short names

The following table lists the short names of the parameters.

Parameter name	Short name
Server	-svr
Username	-usr
Password	-pwd
ServerInstance	-inst
Database	-d
Backup	-bkup

Parameter name	Short name
ArchiveBackup	-arcbk

Examples

Example 1: `delete-backup -d "Db1" -bk "Db1bkup"`

This command deletes the backup set Db1bkup where DB1 is the cloned database.

clone-database

Name

clone-database

Synopsis

This cmdlet enables you to clone a live database or a database that is already backed up in a backup set using the SnapManager SQL Server PowerShell command-line interface.

Detailed description

This cmdlet enables you to clone a live database or a database that is already backed up in a backup set. It creates a backup set of the database and uses the backup set to clone the database. This cmdlet provides you various verification options, DBCC, recovery after restore, retaining backups, management groups and many other options.

You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
clone-database [-Server <ServerName>] [-Username <UserName>]
[-Password <Password>]
[-ServerInstance <comma separated list of original server instances
to restore>]
[-Database <<SQL server instance>,<Number of
databases>,<db1>,<db2>,db3>..>] [-verify]
[-VerDestVolume] [-VerifyOnDestVolumes <Source storage system,
Source volume, Destination storage system, Destination volume>]
[-VerifyServerInstance <SQLServerInstance>]
[-VerSvrLogin <UserName>] [-VerSvrPassword <Password>]
[-TargetDatabase <comma-separated list of database names>]
[-TargetServerInstance <SQL Server Instance>]
[-VerifyDisable] [-RecoverDatabase]
```

```
[-Logbkup <List of transaction logs>]
[-RetainBackups <Number of backups>]
[-MountPointDir <directory pathname>][-UseMountPoint]
[-UpdateMirror][-ManagementGroup <system string
[Daily/weekly/standard]>] [-TruncateLogs] [-AttachDB]
[-NoRetainUTM]
[-RunDBCCAfter] [-RunDBCCBefore][-Command]
[-Runcommand <file pathname>]
[-CmdArguments <List of Command Arguments>]
[-CommandServer <ServerName>] [-GenericNaming][-MountAsNewDb]
[-RetainBackupDays <Integer>] [-RestoreArchivedBackup]
[-NoAccessToRemoteBackup] [-ProxyServer <Proxy server name>]
[-ArchiveBackup] [-VerifyArchiveBackup]
[-ArchivedBackupRetention <Daily/hourly/weekly/monthly/unlimited>]
[<CommonParameters]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

```
-svr win-225-161
```

```
-svr sql1
```

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified.

-ServerInstance (String Parameter): This parameter specifies the SQL server instance where the database is backed up originally. SnapManager takes the local computer name as the default server instance.

You can specify multiple server instance names here as a comma-separated list. If multiple databases reside on the same LUN but are owned by different SQL server instances when you backed them up originally, use the following format:

```
-Inst "SQLServerInstance1","SQLServerInstance2"
```

The first database specified in the -Database parameter refers the first server instance in the -ServerInstance parameter, the second database in the -Database parameter refers to the second server instance in the -ServerInstance parameter and so on.

-Database (String Parameter): Use this option to specify the databases that need to be cloned. Use a comma-separated list of strings:

```
-d Database 1, Database 2, Database 3, Database 4,....
```

Multiple database names should be specified only if those databases share a single LUN or multiple LUNs together. For a multiple database restore, all the selected databases should be present in the selected Snapshot copy.

You can not restore a database with a new name if you specify multiple databases. If you want to restore with a new name, restore those databases one by one. In case of restore to alternate location, specify only one database name.

-Verify (Switch Parameter): Use this parameter if you wish to verify the backed up databases and logs.

-VerDestVolume (Switch Parameter): Use this parameter to verify the database on the SnapMirror destination volume. SnapManager sets it to "false" by default.

-VerifyOnDestVolumes (String Parameter): Specify this parameter in order to override the default SnapMirror relationships. Enter the source and destination storage systems and volumes as a comma-separated list. SnapManager sets it to "false" by default.

-VerifyServerInstance (String Parameter): This parameter specifies the separate SQL server that is used to run the Database Consistency Check (DBCC) utility. If you have not specified the -verify parameter, SnapManager ignores this parameter.

The following example illustrates the usage:

```
-verInst win-225-161
```

Here the SQL server instance is the local or remote SQL server instance to verify on. SnapManager takes the configured SQL server instance that is used for verify in client configuration (registry) as the default SQL server instance.

-VerSvrLogin (String Parameter): This parameter specifies that SQL Server authentication is used. If not specified, the default Windows NT Authentication mechanism is used.

-VerSvrPassword (String Parameter): This parameter is used to input the verification server password. SnapManager ignores this parameter if the parameter -VerSvrLogin is not specified.

-TargetDatabase (String Parameter): Use this parameter to restore a database with a new name. The following example illustrates the usage:

```
-tgDb "NewDatabaseName1"," NewDatabaseName2"," NewDatabaseName3"
```

The parameter defines the new database name to which the original database is restored. The old database name is defined at the same position in the -Database parameter.

If no new database name is given, the database is restored to the original database name the database had during backup. If this original name already exists, the name is modified to: originalDbName__clone, or originalDbName__mount.

-TargetServerInstance (String Parameter): This parameter specifies the name of the new SQL server if you want to restore the database to a new SQL server. SnapManager takes the source SQL server instance as the default.

-VerifyDisable (Switch Parameter): This parameter overrides verification and can disable verification even if the database was not verified after backup.

-RecoverDatabase (Boolean Parameter): Use this parameter to recover the database after the restore operation. SnapManager sets its value to "true" by default.

-ForceRestore (Boolean Parameter): Use this parameter to force the restore of a database based on its state. SnapManager sets its value to "true" by default.

-Logbkup (Switch Parameter): Use this option to specify that the transaction logs also need to be backed up after a full backup.

-RetainBackups (Integer): Use this parameter to specify the number of backups to be retained after the delete operation.

-MountPointDir (String Parameter): Use this parameter to specify the mount point directory on which a backup set will be mounted during database verification. Use this parameter with the parameter -UseMountPoint.

-UseMountPoint (Switch Parameter): This parameter specifies that the Snapshot copy must be mounted to an NTFS directory.

During a SnapManager verification operation, Snapshot copies are mounted to the default NTFS directory for database verification. The option is effective when there are no available drive letters to mount the Snapshot copies. It overrides pre-configured SnapManager verification settings.

-UpdateMirror (Switch Parameter): Use this option to update the SnapMirror destination after a backup or verification operation ends, if the operation uses backups that reside on volumes configured as SnapMirror sources.

-ManagementGroup (String Parameter): This parameter denotes the backup or verify operation that SnapManager performs on daily, or weekly, or standard basis. The default management group is standard.

-TruncateLogs (Switch Parameter): This parameter specifies if the transaction logs need to be truncated. Its value is set to "false" by default, and it is applicable only if the backup option LogOnly is enabled.

-AttachDB (Switch Parameter): If the operation includes a database or transaction log verification, use this option when you want to specify that the databases are to be attached after the verification operation completes.

-NoRetainUTM (Switch Parameter): Use this option if you do not want to retain up-to-the-minute restore ability for older backups in other management groups.

-RunDBCCAfter (Switch Parameter): If the operation includes a database backup, use this parameter if you want to verify the live database after the backups are performed.

-RunDBCCBefore (Switch Parameter): If the operation includes a database backup, use this parameter if you want to verify the live database before the backups are performed.

-**Command (Switch Parameter):** This is a switch parameter that indicates run command after operation.

-RunCommand (String Parameter): This parameter runs the specified command after the SnapManager backup or verification operation is complete. It defines the complete path for the command to be run after the backup or verify operation is complete. SnapManager ignores the -RunCommand parameter if you do not specify the parameter -Command and then SnapManager runs the preconfigured command.

-CmdArguments (String Parameter): This parameter contains a string of SnapManager operation-specific information to be passed to your program or script. It is considered only if the parameters -Command and -RunCommand are specified.

-CommandServer (String Parameter): This parameter specifies the server on which the command is to be run after the backup or verify operation is complete. It is considered only if the parameters -Command and -RunCommand are specified.

-MountAsNewDb (Switch Parameter): This parameter creates a clone of the database.

-GenericNaming (Switch Parameter): This parameter specifies that the backups must follow the Generic backup naming convention.

-RetainBackupDays (Integer): Use this parameter to specify the number of days you want to retain the backups for. SnapManager deletes backups older than the specified number of days. The parameters RetainBackups and RetainBackupDays are mutually exclusive and cannot be specified together.

-ProxyServer (String Parameter): This parameter defines the name of the proxy server. Use it in conjunction with the parameter NoAccessToRemoteBackup.

-NoAccessToRemoteBackup (Switch Parameter): This parameter specifies that there is no direct access to the secondary storage system. SnapManager uses the proxy server to access the secondary storage system.

-RestoreArchivedBackup (Switch parameter): Use this parameter to specify the name of the archived backup set.

-ArchiveBackup (Switch Parameter): Use this parameter to archive database to a secondary storage system.

-VerifyArchiveBackup (Switch Parameter): Use this parameter to verify database archived at the secondary storage system.

-ArchivedBackupRetention (String Parameter): Use this parameter to specify whether you want to retain backups at the archived location on a daily, hourly, weekly, monthly or unlimited basis.

Short names

The following table lists the short names of the parameters.

Parameter name	Short name
Server	-svr
Database	-d
Username	-usr

Parameter name	Short name
Password	-pwd
UseMountPoint	-mp
RetainBackups	-rtbackups
Verify	-ver
VerDestVolume	-verdest
VerOnDestVolumes	-ver
ServerInstance	-inst
VerSvrLogin	-verlogin
VerSvrPassword	-verpwd
VerifyServerInstance	-verInst
TargetDatabase	-tgDb
TargetServerInstance	-tgInst
VerifyDisable	-verDis
RecoverDatabase	-recoverdb
Logbkup	-lb
Managementgroup	-mgmt
RetainBackups	-rtbkups
AttachDB	-attdb
TruncateLogs	-trunclog
NoRetainUTM	-noutm
RunDBCCAfter	-dbccaf
RunDBCCBefore	-dbccbf
Command	-cmd
Runcommand	-runcmd
CmdArguments	-cmdargs
Commandserver	-cmdsvr

Parameter name	Short name
GenericNaming	-gen
MountPointDir	-mpdir
UpdateMirror	-updmir
MountAsNewDb	-mountasnew
ForceRestore	-force
RestoreArchivedBackup	-rstarchbkup
NoAccessToRemoteBackup	-noaccessarchivebkup
ProxyServer	-pxy
ArchiveBackup	-arch
VerifyArchiveBackup	-verarch
ArchivedBackupRetention	-archret
RetainBackupDays	-rtdays

Examples

Example 1: `clone-database -Server sql1 -Database "Db1"`

This command clones database Db1 located on SQL Server sql1.

Default Instance:

Example 2: `clone-database -Server win-225-166 -Inst win-225-166 -Database dbtest1 -Verify -verinst  win-225-166  -RecoverDatabase`

This example enables database cloning with a default name for a default instance.

Example 3: `clone-database -Server win-225-166 -Inst win-225-166 -Database dbtest1 -Verify -verinst win-225-166  -TargetDatabase dbtest1_Clone -RecoverDatabase`

This example enables database cloning with a new name for a default instance.

Named Instance:

Example 4: `clone-database -Server win-225-166 -Inst win-225-166\Named -Database dbtest2 -Verify -verinst win-225-166  -RecoverDatabase`

This example enables database cloning with a default name for a named instance.

Example 5: `clone-database -Server win-225-166 -Inst win-225-166\Named -Database dbtest2 -Verify -verinst win-225-166 -TargetDatabase dbtest2_Clone -RecoverDatabase`

This example enables database cloning with a new name for a named instance.

clone-backup

Name

clone-backup

Synopsis

This cmdlet enables you to clone database from an existing backup or archive using the SnapManager SQL Server PowerShell command-line interface.

Detailed description

This cmdlet enables you to clone a live database or a database that is already backed up in a backup set. This cmdlet restores the database from the existing backup set, to clone the database to an alternate temporary writable LUN location for further use.

You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
clone-backup [-Server <server name>]][-Username <UserName>]
[-Password <Password>] [-ServerInstance <comma-separated list of
original server instance to restore>]
[-Database <<SQL server instance>,<Number of
databases>,<db1>,<db2>,db3>..>]
[-PointInTime <comma separated list of date-times>]
[-TargetDatabase <comma-separated list of database names>]
[-TargetServerInstance <comma-separated list of destination server
instances>]
[-RestoreFromUnmanagedMedia] [-SnapInfoDirectory <directory path>]
[-ForceRestore] [-Backup <Name of backup set>]
[-RestoreLastBackup <number of recent backups to find>]
[-RecoverDatabase]
[-TransLogsToApply <comma separated list of transaction logs to
restore>]
```

```
[-RestoreArchivedBackup][-NoAccessToRemoteBackup]
[-ProxyServer <Proxy Server name>] [RestoreArchive]
[<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

```
-svr win-225-161
```

```
-svr sql1
```

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified.

-ServerInstance (String Parameter): This parameter specifies the SQL server instance where the database is backed up originally. SnapManager takes the local computer name as the default server instance.

If multiple databases reside on the same LUN but are owned by different SQL server instances when you backed them up originally, use the following format:

```
-Inst "SQLServerInstance1","SQLServerInstance2"
```

The first database specified in the -Database parameter refers the first server instance in the -ServerInstance parameter, the second database in the -Database parameter refers to the second server instance in the -ServerInstance parameter and so on.

-Database (String Parameter): Use this option to specify the databases that need to be cloned. Use a comma-separated list of strings:

```
-d Database 1, Database 2, Database 3, Database 4,....
```

Multiple database names should be specified only if those databases share a single LUN or multiple LUNs together. For a multiple database restore, all the selected databases should be present in the selected Snapshot copy.

You can not restore a database with a new name if you specify multiple databases. If you want to restore with a new name, restore those databases one by one. In case of restore to alternate location, specify only one database name.

-PointInTime (String Parameter): Use this switch to restore databases until a specific point in time. The format for the point-in-time string is yyyy-mm-ddThh:mm:ss, with time specified in a 24-hour format.

In case of multiple databases you should specify the point-in-time values for every database separated by a comma. The number of values after the parameter name should equal the number of databases selected. The first value will be applied to the first database specified after the –Database parameter, the second value to the second database, and so on. The following example illustrates the usage:

```
-pit 2008-10-22T11:50:00, 2008-11-25T22:50:00
```

Note

The parameter correspondence is one-to-one, that is, the first point-in-time parameter value specified after the parameter -pit is applied to the first database specified in the parameter -Database and the second point-in-time parameter value to second database and so on.
The values should conform to the required PointInTime regular expression.

-TargetDatabase (String Parameter): Use this parameter to restore a database with a new name. The following example illustrates the usage:

```
-tgDb "NewDatabaseName1"," NewDatabaseName2"," NewDatabaseName3"
```

The parameter defines the new database name to which the original database is restored. The old database name is defined at the same position in the -Database parameter.

If no new database name is given, the database is restored to the original database name the database had during backup. If this original name already exists, the name is modified to: originalDbName__clone, or originalDbName__mount.

-TargetServerInstance (String Parameter): This parameter specifies the name of the new SQL server if you want to restore the database to a new SQL server. SnapManager takes the source SQL server instance as the default.

-RestoreFromUnmanagedMedia (Switch Parameter): Use this parameter if you are restoring databases from archived SnapManager backup sets.

-SnapInfoDirectory (String Parameter): Use this parameter to specify the SnapInfo directory path of the archived backup set. Use the parameter only along with the -RestoreFromUnmanagedMedia parameter.

-RecoverDatabase (Boolean Parameter): Use this parameter to recover the database after the restore operation. SnapManager sets its value to "true" by default.

-ForceRestore (Boolean Parameter): Use this parameter to force the restore of a database based on its state. SnapManager sets its value to "true" by default.

-TransLogsToApply (String Parameter): This parameter specifies the list of transactions logs that need to be applied. SnapManager applies all transaction logs of the databases specified in the -Database parameter by default. You can specify the number of transaction logs to be applied for every database mentioned in the -Database parameter. The list of number of transaction logs that have to applied has to be listed in the same sequence as the databases listed in the -Database parameter. For example,

```
restore-backup -svr MACHINE1\INST1 -database db1,db2
-TransLogsToApply 3,7
```

-ProxyServer (String Parameter): This parameter defines the name of the proxy server. Use it in conjunction with the parameter NoAccessToRemoteBackup.

-NoAccessToRemoteBackup (Switch Parameter): This parameter specifies that there is no direct access to the secondary storage system. SnapManager uses the proxy server to access the secondary storage system.

-RestoreArchivedBackup (Switch parameter): Use this parameter to specify the name of the archived backup set.

-Backup (String Parameter): Use this option to specify the name of the backup set. This is a mandatory parameter. The following example illustrates the usage:

```
-bkup sqlsnap__SYMNASQLDEV170_04-11-2007_15.22.27
```

-RestoreLastBackup (Integer): Use this parameter to restore backups without specifying the name. If you try to use the Backup and RestoreLastBackup parameters together, SnapManager ignores the RestoreLastBackup parameter and uses the backup parameter during restore operation. A typical usage example of the restorelastbackup parameter is as follows:

```
restore-backup -restorelastbackup 1 -backup <backup name>
```

Note

If the value for RestoreLastBackup parameter is 0, SnapManager restores the latest backup. If the value is 1, SnapManager restores second-to-latest backup and so on.

RestoreArchive (Switch Parameter): Use this parameter to restore from archives that are already created.

Short names

The following table lists the short names of the parameters..

Parameter Name	Short name
Server	-svr
Username	-usr
Password	-pwd
ServerInstance	-inst
Database	-d
PointInTime	-pit
TargetDatabase	-tgDb
TargetServerInstance	-tgInst
RestoreFromUnmanagedMedia	-rst
SnapInfoDirectory	-rstumm
ForceRestore	-force
RecoverDatabase	-recoverdb
TransLogsToApply	-translogs
RestoreArchivedBackup	-rstarchbkup
NoAccessToRemoteBackup	-noaccessarchivebkup
ProxyServer	-pxy
RestoreLastBackup	-lastBkup
RestoreArchive	-rstarch

Examples

Example 1: `clone-backup -Server win-225-165 -Database DB2 -Inst win-225-165 -Backup sqlsnap__win-225-165_09-06-2008_13.44.51`

This command creates a clone of the specified backup.

Example 2: `clone-backup -Server win-225-165 -Database DB2 -Inst win-225-165 -RestoreLastBackup 0`

This command restores the most recent clone that was created.

delete-clone

Name

delete-clone

Synopsis

This cmdlet helps you delete a cloned database using SnapManager PowerShell command-line interface.

Detailed description

This cmdlet helps you delete a cloned database using SnapManager PowerShell command-line interface. Before deleting a clone, detach the cloned database instance and disconnect the LUNs.

You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
delete-clone [-Server <ServerName>] [-Database <<SQL server
instance>,<Number of databases>,<db1>,<db2>,db3>..>
[-Username <username>] [-Password <pwd>] [-ServerInstance <SQL
Server Instance>] [<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

`-svr win-225-161`

`-svr sql1`

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified.

-ServerInstance (String Parameter): This parameter specifies the SQL server instance where the database is backed up originally. SnapManager takes the local computer name as the default server instance.

-Database (String Parameter): This parameter specifies the list of databases that are separated by commas. If you do not specify the database parameter, the cmdlet backs up all the databases from all the SQL server instances in the host. This is a mandatory parameter.

Short names

The following table lists the short names of the parameters..

Parameter Name	Short name
`Server`	`-svr`
`Username`	`-usr`
`Password`	`-pwd`
`ServerInstance`	`-inst`
`Database`	`-d`

Examples

Example 1: `delete-clone -svr sql1 -d "Db1"`

This command deletes the clone Db1 on Server sql1.

Import-config

Name

Import-config

Synopsis

This cmdlet enables you to import the configuration information from a SnapManager for SQL control-file using SnapManager PowerShell command-line interface.

Detailed description

This cmdlet enables you to import the configuration information from a SnapManager for SQL control-file using SnapManager PowerShell command-line interface. You can import sections like storage, notification, verification, report, backup, scheduled job, snapmirror volume and so on. You can also control DBCC integrity verification and update statistics table using this cmdlet.

You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information on common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
Import-config [-Server <ServerName>] [-ControlFilePath <filename
and directory path>] [-Section <comma-separated section name list>]
[-Allowlocal]
[-ValidateAndApply] [-Username <username>] [-Password <password>]
[- DBCCBefore] [-DBCCAfter]
[-DeleteOriginalDBFile] [-UpdateStatisticsTable]
[<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

```
-svr win-225-161
```

```
-svr sql1
```

-ControlFilePath (String Parameter) : This parameter specifies the name of the control-file and its path. SnapManager takes the current directory as the control-file path by default.

-Section (String Parameter): This parameter lists section names that are to be imported (separated by commas). If you do not specify any particular section, the default value of all sections is applied. The valid section names that can be applied are as follows: storage, notification, verification, report, backup, scheduledjob, snapmirrorvolume.

-AllowLocal (Switch Parameter): This parameter specifies that the migration of databases to the local disk is permitted. Its value is set to "false" by default.

-ValidateAndApply (Switch Parameter): This parameter specifies whether to apply the imported storage and notification settings data to the current system after validation. If you specify this parameter and validation is successful the imported data will be applied. If you do not specify this parameter only validation occurs.

-**Username (String Parameter):** This parameter denotes the SQL Server account name. If the login name is not specified, SnapManager uses Windows NT Authentication. This parameter is mandatory if you import a scheduled job.

-Password (String Parameter): This parameter is the SQL server account password. SnapManager ignores this parameter if the parameter -UserName is not specified. This parameter is mandatory if you import a scheduled job.

-DBCCBefore (Switch Parameter): This parameter runs the DBCC physical integrity verification before migration. Its value is set to "true" by default.

-DBCCAfter (Switch Parameter): This parameter runs the DBCC physical integrity verification after migration. Its value is set to "false" by default.

-DeleteOriginalDBFile (Switch Parameter): This parameter deletes the copy of the migrated database at original location. Its value is set to "true" by default.

-UpdateStatisticsTable (Switch Parameter): This parameter runs “Update statistics” on tables before detaching the databases. Its value is set to "true" by default.

Short names

The following table lists the short names of the parameters..

Parameter Name	Short name
Server	-svr
ControlFilePath	-config
Section	-sect
ValidateAndApply	-apply
AllowLocal	-tolocal
Username	-usr
Password	-pwd
DBCCAfter	-dbcc2
DBCCBefore	-dbcc
DeleteOriginalDBFile	-deletedbfile
UpdateStatisticsTable	-updatestatistics

Examples

Example 1: `import-config -server "sql1" -ControlFilePath "C:\Program Files\NetApp\SnapManager for SQL\SMSQLConfig_01_23_2007_23.10.20.xml" -Section backup`

This cmdlet validates the backup settings in the control-file. It does not apply the settings to the SQL server.

Example 2: `import-config -Server win-225-166 -Section storage,notification -ControlFilePath "C:\Program Files\NetApp\SnapManager for SQL Server\SMSQLConfig_16July_test4.xml" -ValidateAndApply –AllowLocal`

This cmdlet validates the imported storage and notification settings from control-file and applies it to the system.

Export-config

Name

Export-config

Synopsis

This cmdlet enables you to export the existing configuration information of a SQL server to a control-file using SnapManager Powershell command-line interface.

Detailed description

This cmdlet enables you to export the existing configuration information of a SQL server to a control-file using SnapManager Powershell command-line interface.

You can also implement these options with the SnapManager user interface.

This command also supports the common parameters:

`-Debug (-db), -ErrorAction (-ea), -ErrorVariable (-ev)`

`-OutBuffer (-ob), -OutVariable (-ov), and -Verbose (-vb).`

For more information about common parameters, see

`help about_ubiquitous_parameters.`

Syntax

```
Export-config [-Server <ServerName>] [-ControlFilePath <file
pathname>] [-Section <comma-separated list of section names>]
[<CommonParameters>]
```

Parameters

-Server (String Parameter) : This parameter denotes the name of the host SQL server on which the SQL server instances reside. SnapManager takes the local computer name as the default server name.

Using this parameter, you can also specify a particular SQL server instance. The following examples illustrate the usage:

`-svr win-225-161`

`-svr sql1`

-ControlFilePath (String Parameter): This parameter specifies the name of the control-file and its path. SnapManager takes the current directory as the control-file path by default.

-Section (String Parameter): This parameter lists section names that are to be imported (separated by commas). If you do not specify any particular section, the default value of all sections is applied. The valid section names that can be applied are as follows: storage, notification, verification, report, backup, scheduledjob, snapmirrorvolume.

Short names

The following table lists the short names of the parameters.

Parameter Name	Short name
`Server`	`-svr`
`ControlFilePath`	`-config`
`Section`	`-sect`

Examples

Example 1:
```
export-config -Server win-225-166 -ControlFilePath
"C:\Program Files\NetApp\SnapManager for SQL
Server\SMSQLConfig_16July_test4.xml" -Section storage,notification
```

This cmdlet exports all sections of the existing configuration and settings to the specified control-file.

Configuring SnapManager Application Settings C

About this chapter

This chapter describes SnapManager application settings that can be configured or changed at any time after SnapManager has been installed. The following topics are covered:

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 126
- Appendix A, "Tools for Managing Backup and Verification," on page 359

Overview of SnapManager application settings

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]	◆ Actions pane ◆ Action menu ◆ Configuration Wizard
SnapManager server identity	◆ Add Servers to be Managed ◆ Configuration Wizard
Backup verification settings[2]	◆ Backup Wizard ◆ Action menu ◆ Configuration Wizard
Backup settings	◆ Action menu ◆ Backup Wizard ◆ Backup and Verify option
Clone settings	◆ Action menu ◆ Clone Wizard ◆ Clone option
Restore Settings	◆ Action menu ◆ Actions pane ◆ Restore Wizard ◆ Restore option
Fractional Space Reservation settings	◆ Actions pane ◆ Action menu

Application setting	How the setting can be accessed
Notification settings	◆ Actions pane ◆ Action menu
Run Command After Operation[3]	◆ Action menu ◆ Within the context of a backup or database verification operation: ❖ Backup Wizard ❖ Backup and Verify option
Report directory	◆ Actions pane ◆ Action menu
License settings	◆ Actions pane ◆ Action menu
Server connection settings	◆ Actions pane ◆ Action menu
Find Backups	◆ Actions pane ◆ Action menu ◆ Restore Wizard

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 after installation," on page 79.
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 settings" on page 432. 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.

Connecting to a SQL Server instance

About this section

If you want to add a server that makes all SQL server instances running on the server visible, use the "Add Servers to be Managed" option. SnapManager enables you to connect to the SQL Server you want to manage and to specify the security authentication method to be used to establish the connection. This section contains the following topics:

About the 'Add Servers to be Managed' option

Use the 'Add Servers to be Managed' option to connect to add a server that makes all SQL server instances running on the server visible. When you specify or change the settings, SnapManager immediately connects to the specified SQL Server using the specified security authentication method. These settings remain in effect as the defaults until or unless you change them.

When you start SnapManager, the Add server to be Managed dialog box opens automatically if a *default SQL Server* has not yet been specified. You cannot proceed to use SnapManager until you have successfully added a server to be managed. 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 another server, select "Add Servers to be Managed" to connect to the server running the server account that belongs to the Administrators group on that machine. For more information, see "Connecting to a different SQL server" on page 419.

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:

- SQL Server Agent running as a cluster resource

- SQL Server Agent running on a remote computer
- 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.

Connecting to a different SQL server

To connect to a different SQL Server, complete the following steps.

Note
This changes only the SnapManager server identity on the current machine and does not change the SnapManager server identity on the remote host.

Step	Action
1	If SnapManager is already connected to the default SQL Server, click Disconnect Server in the Actions pane.
2	In the Actions pane, click Add Servers to be Managed. **Result:** The "Add Servers to be managed" window appears.
3	In the "Add Servers to be managed" window, select a server from the list. In a clustered configuration, you can add only a virtual server using the option "Add servers to be managed". **Note** If you want to select an server that is not listed in the "Add Servers to be managed" window, click Browse to add another server, or type the name of the server yourself.

Step	Action
4	Enter the Windows authencitation or SQL server authencitation under Login Details. If you select 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	Click "Add."
6	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.** See "Windows host system requirements" on page 27 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.

Database integrity verification options

About this section

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 421
- "Selecting DBCC options" on page 423
- "Using the Mount Point tab" on page 425

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 37.
2	From the production SQL Server (and not from a remote verification server), open the Verification Settings dialog box using any of these methods: ◆ In the Actions pane, select Backup Verification Settings. ◆ From the Backup Wizard, go to the Verification Settings screen, and then click Verification Settings button. ◆ From the Restore Wizard, go to the Verification Settings screen, and then click Verification Settings button. **Result:** The Verification Settings dialog box appears. The SQL Server option 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 37.
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 select 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 in workgroup mode" on page 35.
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 using any of these methods: ◆ In the Actions pane, select Backup Verification Settings. ◆ From the Backup wizard, go to the Verification Settings screen, and then click Verification Settings button. ◆ From the Restore wizard, go to the Verification Settings screen, and then click Verification Settings button. **Result:** The Verification Settings dialog box appears. The SQL Server option is active by default and displays the host name of the current verification server.
2	Click the DBCC Options tab. **Result:** The DBCC Options option displays the selected DBCC options.
3	In the DBCC Options panel, select 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** `PHYSICAL_ONLY` and `NO_INFOMGS` are selected by default.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>4</td><td colspan="2">By default, the option “Leave database attached after verification” is left unchecked and the database is detached after the DBCC utility finishes.</td></tr>
<tr><td></td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>If you want to detach the database after the verification finishes</td><td>Keep the “Leave database attached after verification” option unchecked.</td></tr>
<tr><td></td><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 Mount Point tab

Use the Mount Point 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 421

To specify which method to use to access database backup Snapshot copies during the database integration verification, complete the following steps.

Step	Action
1	Open the Verification Settings dialog box using any of these methods: ◆ In the Actions pane, select Backup Verification Settings. ◆ From the Backup Wizard, go to the Verification Settings screen, and then click Verification Settings button. ◆ From the Restore Wizard, go to the Verification Settings screen, and then click Verification Settings button. **Result:** The Verification Settings dialog box appears. The SQL Server option is active by default and displays the host name of the current verification server.
2	Click the Mount Point tab. **Result:** The Mount Point displays the currently selected Mount Point options.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>3</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\SnapManager 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>4</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

About this section

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 427
- "Configuring the profile of a transaction log backup" on page 429

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 176.

Step	Action
1	Open the Backup Settings dialog box using any of the following methods: ◆ From the Actions pane, select Backup Settings. ◆ From the Action menu, click Backup Settings. ◆ From the Backup Wizard, go to the Backup Options screen and click Backup Settings. **Result:** The Backup Settings dialog box appears. The Full Database Backup option is active by default and displays the current settings.

Step	Action
2	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. ◆ 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 Naming convention” option. This option is selected by default. For more detailed information, see “SnapManager backup set naming conventions” on page 155.
3	In the “Verify mounted online databases” 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 “Run DBCC physical integrity verification after the backup” option. **Note** By default, both options are not selected because database verification is a time-consuming activity.
4	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 190.

Step	Action
1	Open the Backup Settings dialog box using any of the following methods: ◆ From the Actions pane, select Backup Settings. ◆ From the Action menu, click Backup Settings. ◆ From the Backup Wizard, go to the Backup Options screen and click Backup Settings. **Result:** The Backup Settings dialog box appears. The Full Database Backup option is active by default and displays the current settings.
2	In the dialog box, click the “Transaction Log Backup” tab. **Result:** The Transaction Log Backup tab displays the current settings.
3	In the “Transaction Log Backup” tab, select “Create Snapshot of the SnapInfo drive after backup” to create a Snapshot copy of the snapInfo directory after the backup operation completes.
4	Select “Delete SnapInfo Snapshots” to delete SnapInfo Snapshots based on their number or the number of retention days.
5	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.
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

About this section

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 430
- "Configuring the profile of a restore operation" on page 431

Understanding the restore options

The following table describes each of the Restore Settings options.

Restore option	Description	Default
Leave the database in a state where more logs can be applied	If the database is not fully operational and you want to leave it operational after restore, on selecting this option SnapManager skips the restore and performs the recover operation.	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.	Selected
Retain SQL database replication settings	If this option is selected and you are restoring databases for a 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. Clear this option under the following circumstances: ◆ 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	If this option is selected and the transaction log backup before restore fails, SnapManager aborts the database restore operations. This option is available when the option "Create transaction log backup before restore" is selected.	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 "Preparing to restore an online database as a new database" on page 238.

Step	Action
1	Open the Restore Settings dialog box using any of the following options: ◆ From the Actions pane, select Restore Settings. ◆ From the Action menu, select Restore Settings. ◆ From the Restore Wizard, go to the Restore Settings screen and then click the Restore Settings button. **Result:** The Restore Settings dialog box appears.
2	Select any combination of the restore options you want to use: ◆ Leave the database in a state where more logs can be applied ◆ 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 430.
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 settings

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 complete. 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 complete
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 completed 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.

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 Verify option or the Backup Wizard.

Related topics:

- "Archiving SnapManager backups using a Windows backup utility" on page 332

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 a 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 Verify option: To open the Run Command After Operation dialog box while specifying an operation in the Backup and Verify option, 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 settings" on page 432.

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 settings" on page 432.

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 436.
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 433 and "Command arguments are operation specific" on page 434. **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 3 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.

Enabling or disabling database migration back to local disks

The primary function of the Configuration wizard is to migrate SQL Server databases to 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 “Select a database to move to a lun” screen of the Configuration wizard does not list any local drives unless you explicitly enable the “Enable databases to be migrated to local disk” through Configuration Wizard Option Settings.

Related topics:

- “Understanding control-file based configuration” on page 127
- “Migrating databases to LUNs using the Configuration Wizard” on page 143

To enable or disable the ability to migrate databases from LUNs back to local disks, complete the following steps.

Step	Action	
1	From the Actions pane, click “Configuration Wizard Option Settings”.	
2	The option “Enable databases to be migrated to local disk” appears.	
	If...	**Then...**
	You need to enable database migration back to local disks	Select the option “Enable databases to be migrated to local disk”
	You need to disable database migration back to local disks	Clear the option “Enable databases to be migrated to local disk”.
3	Click OK to close the dialog box.	

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 440
- "Reasons to change the report directory" on page 440
- "Accessing reports created in a previous directory" on page 441
- "Using the Report Directory Setting dialog box" on page 441

Related topics:

- "SnapManager creates a share for remote access to the Report directory." on page 81
- "Managing SnapManager Operational Reports" on page 263

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 option 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.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">From the SnapManager Actions pane, select Report Directory Setting.

Result: The Report Directory dialog box appears and displays the current location of the report directory.</td></tr>
<tr><td>2</td><td colspan="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.

<table><tr><th>If...</th><th>Then...</th></tr><tr><td>You know the full directory path name</td><td>Click in the Report Directory box and modify the path name.</td></tr><tr><td>You prefer to browse to the new location</td><td>Click Browse to use a browse dialog box to select the new location.</td></tr></table></td></tr>
<tr><td>3</td><td colspan="2">To apply your changes and close the Report Directory Setting dialog box, click OK.</td></tr>
<tr><td>4</td><td colspan="2">To refresh the information displayed in the SnapManager Reports option, go the SnapManager Actions pane and select Refresh.</td></tr>
</table>

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 442
- "Using the Auto Notification Settings dialog box" on page 443

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 clone
- 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 Actions pane, select Notification Settings. **Result:** The 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	This selection is optional. To enable e-mail notification, select the "Send e-mail 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	

Step	Action
5	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
6	Click OK to commit your settings.
7	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	
8	If you want to enable posting of SnapManager events to the storage system event log, select the "Log SnapManager events to storage system syslog" option.
9	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.
10	If you want to limit SnapManager event logging to failure events, select the "On failure only" option.
11	Click OK. **Note** If you are using the Configuration Wizard instead of the Auto Notification Settings dialog box, click Next.

Configuring Post-Restore Database Recovery D

Overview

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:

Related topics:

- "Restoring using the SnapManager Restore Wizard" on page 245
- "Restoring using the SnapManager Restore option" on page 241

Understanding post restore database recovery states

Understanding post restore database recovery states

The following table describes the post restore database states from which you can select.

Database state	Description
Operational	All of the following apply: ◆ No more transaction logs can be restored. ◆ The database is ready to use. This database state is selected by default.
Non-Operational	More transaction logs can be restored.
Read-Only	All the following apply: ◆ More transaction logs can be restored. ◆ The undo file is enabled. If more transaction logs are restored, any changes can be rolled back if 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 databases

Overview

When specifying database restore operation, you can select 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 database recovery state from SnapManager Restore" on page 449
- "Specifying database recovery state from within the Restore wizard" on page 451

Related topics:

- "Understanding SnapManager Restore" on page 228
- "Performing a restore operation" on page 237

Specifying database recovery state from SnapManager Restore

When using the SnapManager Restore option to restore multiple databases, you use the SnapManager for SQL Server-Restore 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 database restore operation started using SnapManager Restore, complete the following steps.

Step	Action
Open the SnapManager for SQL Server-Restore dialog box.	
1	In the Actions pane, click Restore. **Result:** The "SnapManager for SQL Server-Restore" dialog box appears and prompts you to select the post-backup state for the databases.
2	Select the databases to be restored from the list that appears. This is described in "Restoring using the SnapManager Restore option" on page 241. As described that procedure, click Restore when you are ready to start the restore operation.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Specify the post-restore database state.</td></tr>
<tr><td>3</td><td>Select the state that you want the database to be left in after the restore operation finishes.
Leave the databases operational. No more transaction logs can be restored.
Leave the databases nonoperational but able to restore more transaction logs.
Leave the databases read-only and able to restore more transaction logs.
For descriptions of the database recovery states, see “Understanding post restore database recovery states” on page 448.<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 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 nonoperational...” option.
2. In the database list in the middle of the dialog box, clear 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>4</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
Start the multiple-database restore operation.	
5	Continue with the procedure described in "Restoring using the SnapManager Restore option" on page 241.

Specifying database recovery state from within the Restore wizard

When using the Restore wizard to restore databases, you use the "Database state after 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 database restore operation, until you reach the "Database state after restore" screen. This is described in "Restoring using the SnapManager Restore Wizard" on page 245. **Result:** The "Database state after restore" screen prompts you to select 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 the databases operational. No more transaction logs can be restored.
Leave the databases nonoperational but able to restore more transaction logs.
Leave the databases read-only and able to restore more transaction logs.
For descriptions of the database recovery states, see “Understanding post restore database recovery states” on page 448.<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 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 nonoperational...” option.
2. In the database list in the middle of the dialog box, clear 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 go to the next wizard screen, click Next. **Result:** The Restore wizard displays the "Restore Database As" screen.
Finish specifying the single-database restore operation.	
5	Continue with the procedure described in "Restoring using the SnapManager Restore option" on page 241.

Managing fractional space reservation

E

About this appendix

This topic summarizes Data ONTAP fractional space reservation and describes how to configure SnapManager fractional space reservation policies:

About fractional space reservation

Overview

The following paragraphs summarize space reservation and fractional space reservation as supported by Data ONTAP 7.1 or greater. For more detailed information about these features, see the *Data ONTAP Block Access Management Guide for iSCSI and FCP* for release 7.1 or later.

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 of a lack of disk space on the storage system. Other operations, such as taking a Snapshot copy or the creation of 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 and the host application may terminate, report I/O errors, or experience unexpected behavior.

Fractional space reservation: With fractional reserve, 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 reserve 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 overwrite reserve amounts are configured at the volume level; that is, 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.

What can happen with a fractional space-reserved volume

Overview

When a LUN is fully space reserved, write operations to that LUN are guaranteed against failure caused by an out-of-space condition due to Snapshot copy disk space consumption. When the overwrite reserve for a volume is set to less than 100 percent, however, write operations to the LUNs on that volume are no longer guaranteed when the storage system volume runs low in free disk space due to Snapshot copy space consumption.

Attention

If a storage system volume runs out of overwrite reserve space, write operations to a LUN on that volume will fail and the host application may terminate, report I/O errors, or exhibit unexpected behavior.

Data ONTAP provides two space management tools to ensure that a fractionally space-reserved volume does not run out of overwrite reserve: automatic FlexVol expansion and automatic Snapshot copy deletion from FlexVol volumes. These features, summarized in the following paragraphs, monitor the reserved space and take action if the free space becomes scarce. For more detailed information, see the *Data ONTAP Block Access Management Guide for iSCSI and FCP* for release 7.1 or later.

Automatic expansion of flexible volumes: Data ONTAP can automatically expand the volume that is used to store Snapshot copy data, provided the volume is a flexible volume with the guarantee option set to `volume`. When the flexible volume is nearly full, Data ONTAP automatically expands the volume into the space preallocated for it in the aggregate. The automatic Snapshot copy deletion and FlexVol expansion features can be enabled separately, or together with one policy to be applied before the other. When fractional-space-reserved volumes hold LUNs that store SQL Server database files, however, only the automatic FlexVol expansion feature can be used, if needed.

Automatic deletion of Snapshot copies from flexible volumes: Data ONTAP can automatically delete one or more Snapshot copies on the volume, provided the Data ONTAP Snapshot copy autodeletion policy is enabled and set to trigger when the overwrite reserve is nearly full on the volume. If the trigger condition is detected, the oldest or newest Snapshot copies are deleted until a configured percentage of the volume is free space. If you do not want to automatically delete SnapShot copies on the volume, you can set the overwrite

reserve to 100 percent, by setting the fractional space reserve to 100 percent on the storage system. Note that this Data ONTAP feature is not designed specifically to support backup and restore operations on SQL Server databases:

- The options for selecting Snapshot copies to be deleted do not have visibility to the automatic backup Snapshot copy deletion criteria configured in SnapManager.
- SQL Server administrators want to retain at least one online backup for each database at all times.

If the Data ONTAP Snapshot copy autodelete policy is enabled for a volume that stores SnapManager backup set components, either disable the policy or configure it so that it does not delete the SnapManager backup set components. For more information, see "Viewing fractional space reservation status" on page 463.

Fractional space reservation policies manage SQL Server data

Overview

In a SnapManager environment in which SQL Server data is stored on LUNs in a fractional space-reserved storage system volume, the SQL Server administrator needs to avoid an out-of-space condition on the volume in a way that allows explicit or implicit SQL Server-aware control over the deletion of SQL Server backup set components. To address this need, SnapManager provides its own space management tool for monitoring overwrite reserve utilization on the volumes. If overwrite reserve space runs low for a fractional space-reserved volume, SnapManager can take action to prevent the overwrite reserve from becoming fully depleted. Specifically, SnapManager can *delete SQL Server backup sets* or *dismount SQL Server databases* (or both), triggered when the overwrite reserve utilization for the volume reaches specific thresholds specified in the fractional space reservation policy.

Note

If SnapManager e-mail notification is enabled, SnapManager sends SMTP e-mail after a SnapManager fractional space reservation policy event finishes.

Automatic deletion of SQL Server backups: SnapManager provides for the automatic deletion of backups of LUNs that store SQL Server data. When enabled, this component of the SnapManager fractional reservation policy serves as the SQL Server-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 Server backups as follows:

- Delete the oldest backup Snapshot copies first.
- Retain the specified number total backup Snapshot copies on the volume.
- Retain the most recent backup of any database (if it resides on the volume).
- Retain any backups of databases no longer in existence.

Select the backup retention level based on 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 Server-aware features, the automatic deletion of Snapshot copies does not necessarily prevent an out-of-space condition on the volume.

SnapManager execute based on the policy for the volume that exceeds the thresholds, not other volumes that could exist in the same backup set.

For example, suppose you have an SQL Server the has backup that span multiple volumes and with the following automatic deletion thresholds configured:

- Volume 1: Delete all but 2 Snapshot copies if 20% overwrite reserve utilization is exceeded.
- Volume 2: Delete all but 5 Snapshot copies if 20% overwrite reserve utilization is exceeded.
- Volume 3: Delete all but 10 Snapshot copies if 20% overwrite reserve utilization R is exceeded.

If the 20% overwrite reserve utilization threshold for Volume 1 is exceeded, SnapManager deletes all but two Snapshot copies, regardless of the policies for Volumes 2 and 3. If the 20% overwrite reserve utilization threshold for Volume 2 is exceeded, SnapManager deletes all but five Snapshot copies., regardless of the policies for Volumes 1 and 3.

Set the same number of backup sets to delete on SQL Server database and transaction logs LUN residing on storage system volumes. If there is a mismatch in this number, SnapManager attempts to delete backup sets based on the fractional reserve policy settings.

Automatic dismounting of SQL Server databases: SnapManager provides for the automatic dismounting of SQL Server databases in space-reserved LUNs, triggered if overwrite reserve utilization on the volume reaches the threshold specified by the fractional space reservation policy. This effectively stops SQL Server write operations to LUNs in a storage system volume where overwrite reserve space is nearly full. This second component of the fractional space reservation policy is a last resort action that prevents further consumption of overwrite reserve. Therefore, it is always enabled.

When both components of a fractional space reservation policy are enabled, the *dismounting of SQL Server databases* must be triggered at a *later level of overwrite reserve utilization* than is used to trigger the *deletion of SQL Server backup Snapshot copies*. This causes SnapManager to first use backup set deletion to free up 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 may still run out and the storage system volume will go offline. For this reason, it is recommended that dedicated volumes are used for SQL Server 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
Deleting backup Snapshot copies of SQL Server		
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 Server databases		
Status:	Always enabled[1]	
Trigger on overwrite reserve utilization:	90%	1% – 99% [2]

[1] Enabling automatic deletion of backup Snapshot copies of SQL Server does not necessarily prevent an out-of-space condition on the volume. Therefore, SnapManager always enables database dismounting.

[2] Enabling automatic deletion of backup Snapshot copies of SQL Server 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 before 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 reserve set to less than 100 percent. It should also contain LUNs that store SQL Server database files, SQL Server transaction log files, or SnapManager SnapInfo directories.

Default policy with defaults: 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: Optionally, you can customize the default policy that is applied to all storage system volume that contains fractional space-reserved LUNs.

Volume-specific policies: Optionally, you can override the default policy for any particular volume that contains fractional-space-reserved LUNs, by applying a custom policy.

Viewing fractional space reservation status

Viewing fractional space reservation status

In the Fractional Space Reservation Settings dialog box, use the Current Settings tab to view the current space consumption in the storage system volumes that contain LUNs that store SQL Server data or SnapInfo directories.

Drive Letter or Mountpoint: *A SnapManager configuration setting for the LUN.* The drive letter or NTFS mountpoint on which the LUN is mounted.

Fractional Reserve (%): The amount of space reserved for overwrites on the storage system volume that contains this LUN. 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.* The percentage of overwrite reserve utilization that triggers automatic deletion of SQL Server backup sets.

Disable Database Trigger (%): *A SnapManager fractional space reservation policy setting for the storage system volume that contains the LUN.* The percentage of overwrite reserve utilization that triggers automatic disabling of SQL Server databases.

Used Reserve: For the storage system volume that contains this LUN, the amount of overwrite reserve *in use*. Expressed in two ways: as a percentage of the total size of all space-reserved LUNs in the volume and in megabytes.

Available Reserve (MB): For the storage system volume that contains this LUN, the amount of overwrite reserve *available*.

Snapshot Autodelete: For the storage system volume that contains this LUN, the state of the Data ONTAP Snapshot copy autodeletion feature: enabled or disabled. If this LUN stores SQL Server 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	Select Fractional Space Reservation Settings in the SnapManager Actions pane.
2	In the Current Status tab, note the space consumption status for each LUN that stores SQL Server data or SnapInfo directories. ◆ The following columns displays SnapManager configuration information: ❖ Drive Letter or Mount Point ❖ Fractional Overwrite Reserve(%) ❖ 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 displays the fractional overwrite reserve settings and status: ❖ Used Overwrite Reserve (% ❖ Used Overwrite Reserve (MB) ❖ Used 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 is automatically refreshed 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 Server data or SnapManager SnapInfo directories are stored on LUNs contained in a volume with these characteristics, the Data ONTAP Snapshot copy autodeletion policy might delete SQL Server backup set components. 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 Server 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 Server-aware autodeletion feature. For details, see "Automatic deletion of SQL Server backups" on page 459 and "Configuring fractional space reservation policies" on page 466. The SnapManager autodeletion feature can be used in place of or in conjunction with the Data ONTAP autodeletion feature; you can also select to disable the SnapManager autodeletion feature.
4	To close the dialog box, click OK.

Configuring fractional space reservation policies

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 "About the default fractional space reservation policy" on page 462.

Note

SnapManager automatically applies the default policy to every storage system volume that contains fractional-space-reserved LUNs that store SQL Server 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	Select Fractional Space Reservation Settings in the SnapManager Actions pane. **Result:** The Fractional Space Reservation Settings window is displayed.
2	Select the Policy Settings tab.

Step	Action	
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.
Enable or disable fractional space reservation monitoring		
4	**If you want to...**	**Then do this...**
	Enable fractional space reservation monitoring	Select the Enable Fractional Space Reservation Monitoring check box.
	Disable fractional space reservation monitoring	Clear the Enable Fractional Space Reservation Monitoring check box.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td colspan="2">Disable or configure automatic deletion of SQL Server backup Snapshot copies</td></tr>
<tr><td>5</td><td>Use the “Automatic deletion of backups” panel to disable, enable, or configure automatic deletion of SQL Server backup Snapshot copies in fractional-space-reserved LUNs on the volume.

Note
Although automatic deletion of SQL Server 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 Server data.

For more information, see “Automatic deletion of SQL Server backups” on page 459.

<table><tr><th>If you want to...</th><th>Then do this...</th></tr><tr><td>Enable automatic deletion of SQL Server backup Snapshot copies</td><td>Select the “Delete backups that include LUNs which have less than 100% overwrite reservation” option, and then skip ahead to Step 8.</td></tr><tr><td>Disable automatic deletion of SQL Server backup Snapshot copies</td><td>Clear the “Delete backups that include LUNs which have less than 100% overwrite reservation” option, and then proceed to Step 6.</td></tr></table>Note
Data ONTAP includes a separate Snapshot copy autodeletion feature. For details, see “Viewing fractional space reservation status” on page 463. The SnapManager autodeletion feature can be used in place of or in conjunction with the Data ONTAP autodeletion feature.</td></tr>
</table>

Step	Action
6	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 Server 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 “Automatic dismount of databases” panel.
7	In the “Number of most recent backups to retain” field, enter the number of backups 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 Server backups” on page 459.

Step	Action
Configure automatic dismounting of SQL Server databases	
8	Use the “Automatically dismount databases” panel to configure automatic dismounting of SQL Server databases in fraction-space-reserved LUNs on the volume. **Note** Because automatic deletion of SQL Server 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 Server 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 later 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 Server databases” on page 460.
Apply the changes to the default or volume-specific policy	
9	To apply your changes and close the dialog box, click OK.

Index

Symbols

A

B

G

H

I

L

M

N

O

P

R

S

T

U

V

W