Technical Report

SnapManager for SQL Deployment Guide

Abhishek Basu, Chittur V. Narayankumar, NetApp
October 2011 | TR-3941

TABLE OF CONTENTS

1 INTRODUCTION 4

1.1 PREREQUISITES 4

1.2 SCOPE 4

1.3 ACCESS METHODS TO SHARED STORAGE 5

2 STORAGE LAYOUT 5

2.1 STORAGE LAYOUT CONSTRAINTS 5

2.2 LAYOUT OF MICROSOFT SQL SERVER INSTANCE 8

2.3 LAYOUT FOR SMALL DATABASES (<80GB SIZE) 9

2.4 LAYOUT FOR LARGE DATABASES (>500GB <1TB SIZE) 10

2.5 SUPPORT FOR VERY I/O-INTENSIVE DATABASES 11

3 CONFIGURATION RULES 11

3.1 WINDOWS CONFIGURATION RULES 11

3.2 MICROSOFT SQL SERVER CONFIGURATION RULES 12

3.3 SNAPMANAGER FOR SQL CONFIGURATION RULES 12

3.4 SNAPMANAGER FOR SQL—ADDITIONAL INFORMATION 18

3.5 RUN THE MICROSOFT SQL SERVER CONFIGURATION WIZARD 21

3.6 CONFIGURATION WIZARD 22

4 DATABASE VERIFICATION 31

4.1 DATABASE CONSISTENCY CHECK—DATABASE VERIFICATIONS USING SMSQL 31

5 CONFIGURATION OF CONTROL-M SCHEDULING 34

6 APPENDIXES 34

6.1 DEFINITIONS 34

6.2 PREREQUISITES FOR INSTALLING SNAPMANAGER FOR SQL 35

LIST OF TABLES

Table 1) Supported configurations: Microsoft SQL Server edition. 4

Table 2) Required Windows hotfixes. 4

Table 3) Access method assumptions. 5

Table 4) Microsoft SQL Server instances with LUN layout. 9

Table 5) Multiple replication and recovery options. 12

Table 6) Multiple replications with scheduled details. 13

Table 7) Replication operation with scheduling details. 13

Table 8) SnapManager for SQL configuration limits. 14

Table 9) AD groups with different environments. 21

Table 10) SnapManager for SQL Server configuration wizard. 23

Table 11) Database verification. 32
Table 12) Drive size with path. 33
Table 13) Control-M scheduling. 34
Table 14) Glossary of terms. 34
Table 15) Configuring SMSQL on VMDK. 36
Table 16) Changing the NFS.MaxVolumes setting. 43
Table 17) Installing SMSQL on a dedicated Windows server. 44
Table 18) Installing SMSQL in unattended mode. 46
Table 19) Restore option descriptions. 48
Table 20) Configuring SMSQL notification settings. 50

LIST OF FIGURES

Figure 1) SMSQL design rules. 6
Figure 2) Storage layout best practice. 7
Figure 3) LUN and drive layout for an instance. 8
Figure 4) Volume mount-point drive. 9
Figure 5) LUN and drive layout for a small user database. 10
Figure 6) LUN and drive layout for a large user database. 11
Figure 7) Host backup software installation. 15
Figure 8) SMSQL backup process. 17
Figure 9) Different cloning options. 18
Figure 10) Databases to be cloned. 19
Figure 11) Naming the cloned database. 19
Figure 12) Clone settings. 20
Figure 13) Querying the cloned database. 20
Figure 14) Microsoft SQL Server in SnapManager. 30
Figure 15) Microsoft SQL Servers. 30
Figure 16) SnapManager for SQL Server. 31
Figure 17) SnapManager for SQL restore setting. 48
Figure 18) SnapManager report directory setting. 49

1 INTRODUCTION

This implementation guide covers best practices to set up a NetApp® SnapManager®-enabled Microsoft® SQL Server® according to the customer's standards for network-attached storage.

Use Table 1 to record the SQL Server configurations for which SnapManager is to be deployed.

Table 1) Supported configurations: Microsoft SQL Server edition.

Microsoft SQL Server	Windows 2008 Enterprise Edition x64	Windows 2008 Standard Edition x64

See the NetApp Interoperability Matrix at http://now.netapp.com/matrix for exact supported service package versions.

To install and configure SnapManager for SQL Server (SMSQL), you must have:

- NetApp SnapManager for SQL Server
- NetApp SnapDrive® for Windows®
- Microsoft .Net Framework 3.5 SP1
- Microsoft PowerShell 1.0 RTM x64 Edition
- SnapManager service account with system administrator rights on the host running the Microsoft SQL Server databases
- The Snap Drive service account must have administrative rights on the host.

Record the required hotfixes for the installation of the operating system in Table 2.

Table 2) Required Windows hotfixes.

OS	Hotfixes
Windows 2008 Enterprise Edition x64	
Windows 2008 Standard Edition x64	

1.1 PREREQUISITES

The following are required before proceeding with the instructions in this document:

- A supported Standard Operating Environment (SOE) Windows build
- Installed and connected cables
- Configured storage switching infrastructure

1.2 SCOPE

This document covers the following areas:

- Microsoft SQL Server 2008 deployments using iSCSI and NFS storage
- Volume and qtree layout and configuration for Microsoft SQL Server database deployments
  - Small: database size <80GB
  - Large: database size <1TB
- Naming conventions pertaining to the Microsoft SQL Server storage design and activation
- Mount-point layout and configuration

- Microsoft SQL Server database configuration recommendations consistent with NetApp and Microsoft best practices for Microsoft SQL Server deployments utilizing iSCSI storage

This document does not cover the following areas:

- Microsoft SQL Server deployments using Common Internet File System (CIFS) storage
- Detailed step-by-step procedures for Microsoft SQL Server installation, configuration, and customization

1.3 ACCESS METHODS TO SHARED STORAGE

The clustered NetApp storage controllers are accessed through a redundant Gigabit Ethernet network or, in some cases, through Fibre Channel Protocol (FCP).

NetApp assumes that storage is accessed utilizing the protocols, versions, and configuration details shown in Table 3.

Table 3) Access method assumptions.

Access Method	Assumption or Constraint
Ethernet SAN iSCSI	Ethernet SAN iSCSI network redundancy is provided by Microsoft MPIO multipathing support for iSCSI (MSMPIO), which is part of the Microsoft iSCSI initiator. MSMPIO provides an active/active resilience virtual link over two physical links.
FCP SAN	FCP-SAN redundancy is provided by the Data ONTAP® DSM 3.1 for Windows MPIO (DSM-MPIO).

2 STORAGE LAYOUT

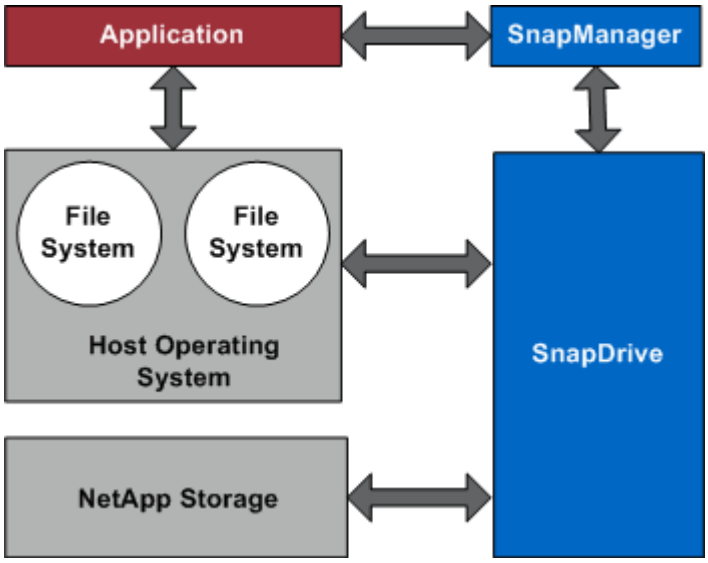
This section describes the best practices for storage layout design for Microsoft SQL Server databases on top of NetApp storage with SnapManager.

2.1 STORAGE LAYOUT CONSTRAINTS

SnapManager for SQL Server (SMSQL) adds constraints to the way in which databases are placed on LUNs. This section describes how Microsoft SQL Server databases are laid out on NetApp storage.

SMSQL is a Windows host-based application that manages rapid and operational backups and recoveries for Microsoft SQL Server databases. SMSQL interacts with SnapDrive for Windows to control the Storage LUNs and with Microsoft SQL Server so that consistent Snapshot™-based database backups are created as shown in Figure 1.

Figure 1) SMSQL design rules.



NetApp SnapManager for SQL uses Windows VSS Framework to create rapid application-consistent backups. This technology interfaces with the NetApp Snapshot process. Backups are initiated by SMSQL when an on-the-fly or a scheduled backup is requested. SMSQL will communicate with the VSS Framework to verify the health of the VSS Writer and, if all is well, the SQL databases will be placed into a frozen state. The freeze is scheduled to end after 10 seconds and is hard-coded into Microsoft VSS. During the freeze, SMSQL will call SnapDrive for Windows (SDW) to request that a NetApp Snapshot copy be created on the NetApp controller. Once the NetApp Snapshot copy has been created, SMSQL will notify Microsoft VSS that the backup is complete and the freeze should be released. SMSQL will then notify Microsoft SQL Server that a full backup has completed. It also supports taking streaming backups (normal Microsoft SQL Server backups) of transaction log files and of the master database.

The following is a summary of best practice recommendations:

- The Microsoft SQL Server sample databases (Pubs, Northwind, Adventureworks, and so on) must be excluded from all Microsoft SQL Server installation, unless they are intended to be used as user databases for development purposes.
- Each Microsoft SQL Server instance must have its Master, Msdb, and Model database and log files on the same dedicated iSCSI LUN. This LUN resides on a dedicated FlexVol® volume for each instance within a vFiler® unit.
- Each Microsoft SQL Server instance must have the Tempdb[1] database on a dedicated iSCSI LUN on a dedicated or shared FlexVol volume. Since this data is transient in nature, it does not require backup or replication support. These LUNs and FlexVol volumes must never be replicated with SnapMirror®. Ideally, this FlexVol volume must be located on a separate aggregate from all other databases.
- Each Microsoft SQL Server instance must have a separate Snapinfo iSCSI LUN created. This contains stream-based (non-Snapshot-based) backups of system databases, backups of transaction

[1] TEMPDB is recreated every time SQL Server is restarted and provides a scratch pad space in which SQL Server manages the creation of indexes and various sorting activities. To support these activities some TEMPDB files become quite large.

logs, and metadata for all backups per Microsoft SQL Server instance. A dedicated FlexVol volume is created per Microsoft SQL Server instance for the Snapinfo directory.

- User databases are located on one or more iSCSI LUNs. Multiple databases do not occupy one iSCSI LUN, although this might be required under certain circumstances.
- For large and very large databases:

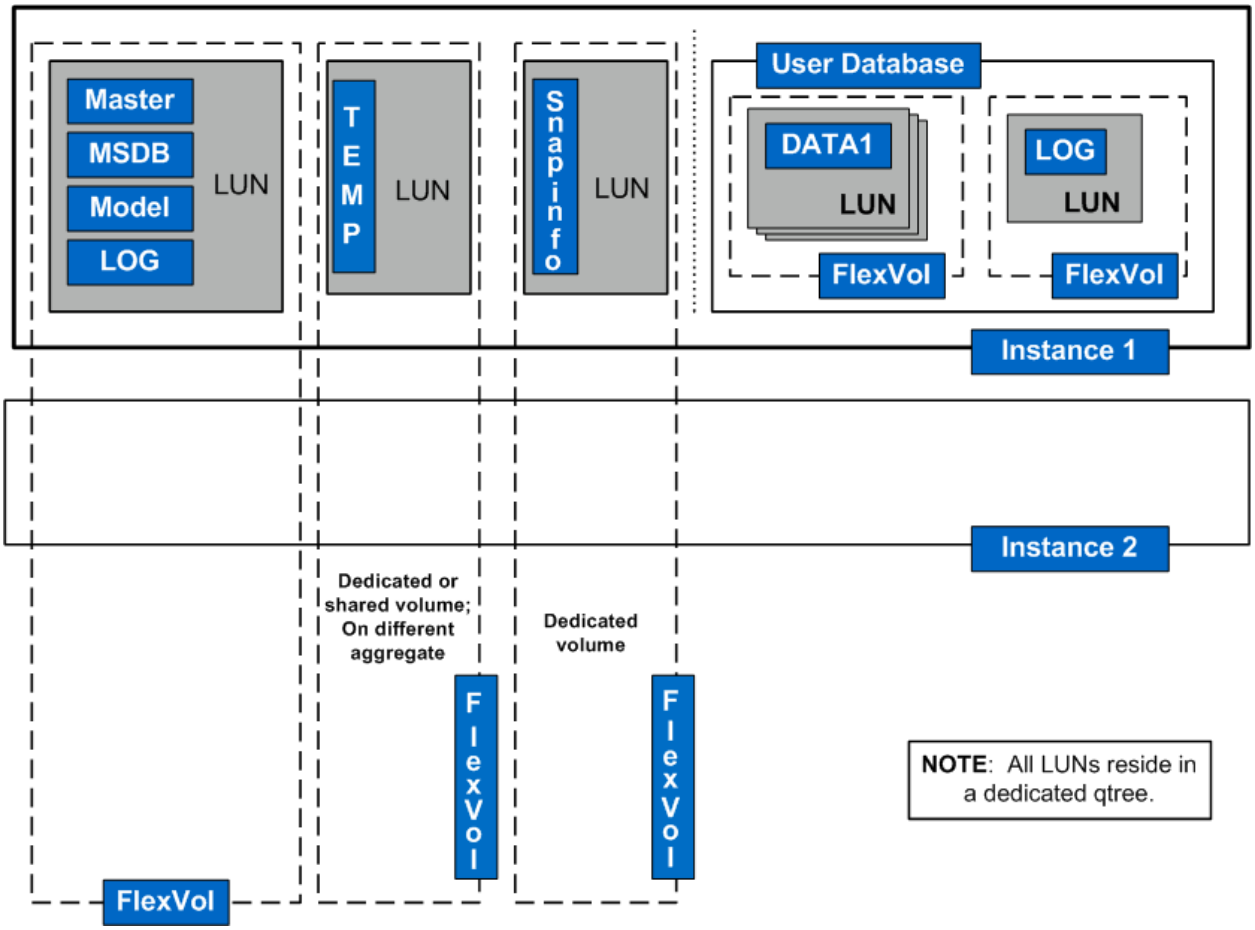
  i) Database files are stored in multiple LUNs in a dedicated FlexVol volume and their associated transaction logs are placed in the dedicated LUN in a separate FlexVol volume.

  ii)To have separate LUNs for Logs and Databases and to group data files and log files across the two LUNs.
- If multiple database files share a LUN, you can restore an individual database but be aware that a lun clone split restore will not occur. Instead the data will be copied from a mounted snapshot back into the active file system.
- Database and log files must never be located on a mount point root LUN because this can cause SnapManager for SQL backups and restores to fail.

Figure 2 depicts the LUN layout resulting from listed best practices. The layout can be separated into two parts:

- The layout of the Microsoft SQL Server instance (left part): This layout is independent of the user databases keeping custom data.
- The layout of the user databases (right part): This layout depends on the user database size and its utilization. Several user databases might exist and be served by one Microsoft SQL Server instance.

Figure 2) Storage layout best practice.



2.2 LAYOUT OF MICROSOFT SQL SERVER INSTANCE

The storage layout of the Microsoft SQL Server instance uses three FlexVol volumes. Each FlexVol volume provides a LUN encapsulated in its own qtree. All LUNs are thin-provisioned.

- FlexVol volume qtree with LUN for: Master, Msdb, Model, Log
- FlexVol volume with qtree with LUN for: Tempdb
- FlexVol volume with qtree with LUN for: Snapinfo

The size of the LUN for the Master, Msdb, and Model database is never changed, but the size of the Tempdb LUN can be dynamically increased through SnapDrive. A Microsoft SQL Server instance does not need to be taken offline to extend a LUN with SnapDrive.

Depending on the required size of the user database, additional LUNs must be provided as described in the following sections.

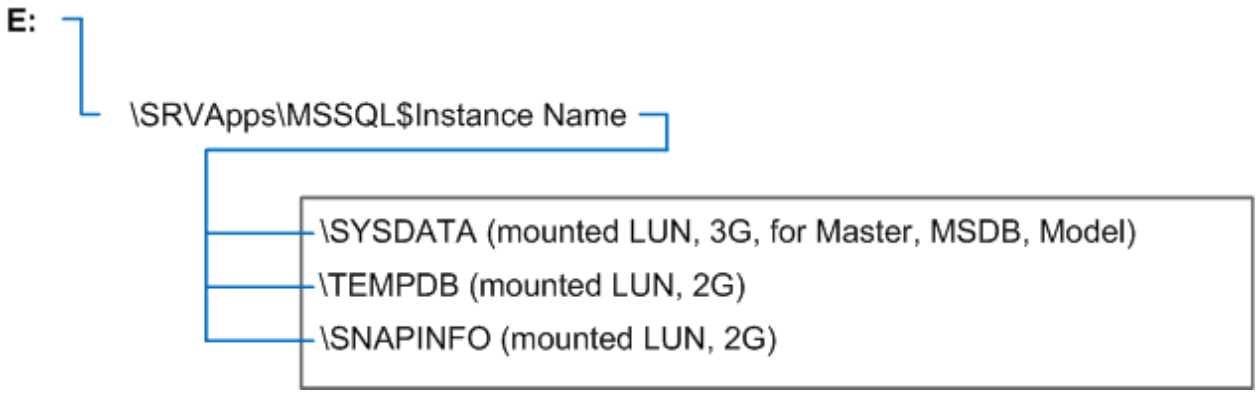
LUN AND DRIVE LAYOUT

NetApp recommends the following LUN and drive layout, which provides a handy abstraction for management purposes:

1. Map the data of each Microsoft SQL Server instance on this server to a drive using the folder structure provided.
2. Organize the data of the Microsoft SQL Server instance and the user databases served by this instance as mount points.

Figure 3 shows the directory hierarchy and the mount points.

Figure 3) LUN and drive layout for an instance.

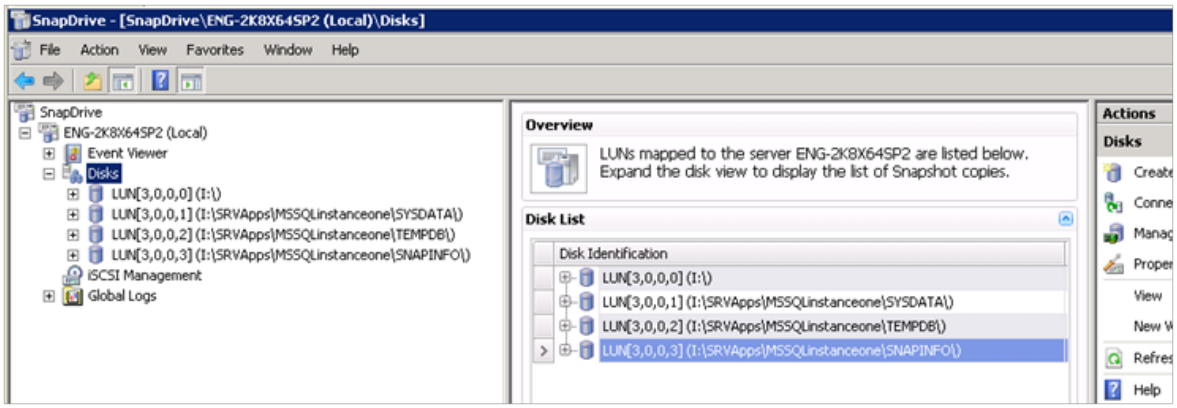


Step-by-step instructions for each Microsoft SQL Server instance:

1. Create a thin-provisioned LUN that is 3GB in size. Map it to drive letter I: for Microsoft SQL Server instance 1 (J: for instance 2, and so on).
2. Create the directory \SRVApps\MSSQL$InstanceName. MSSQL$InstanceName is the Microsoft SQL Server instance name.
3. Mount the LUNs to the mount points sysdata, tempdb, and Snapinfo under the created directory structure.

Figure 4 shows the drive and LUN layout for a Microsoft SQL Server instance named *instanceone*. This view is generated by the SnapDrive MMC plug-in.

Figure 4) Volume mount-point drive.



RUNNING MULTIPLE INSTANCES

Multiple Microsoft SQL Server instances are supported by SnapManager. To add additional instances, reserve the LUNs and map/mount them in the same way as described earlier. The FlexVol volumes created for the first Microsoft SQL Server instance can be shared between multiple instances.

Following are best practices:

- Database files cannot reside on a LUN that is the root of a mount point (directly under the drive letter or a LUN).
- If a drive is marked with MPRoot (mount point root) in the SMSQL graphical user interface, the database cannot be migrated to it.
- The Snapinfo directory can reside on a volume mount point, as with a standard volume.
- Backup, restore, and other SMSQL operations are no different between mounted volumes and standard volumes; mounted volumes just have longer paths.

Table 4) Microsoft SQL Server instances with LUN layout.

Microsoft SQL Server Content	LUN Layout	FlexVol	Aggregate
Master, Msdb, and Model database	Dedicated	Shared between instances	No preference
Tempdb database	Dedicated	Shared between instances	Separate from database and logs
Snapinfo directory (one per Instance)	Dedicated	Shared between instances	No preference

2.3 LAYOUT FOR SMALL DATABASES (<80GB SIZE)

This section describes the layout for one or more user databases up to 80GB in size. NetApp recommends that you store the DATA and LOG of a user database in two separate LUNs.

For each user database, two additional FlexVol volumes are used. Each FlexVol volume provides a qtree with a LUN in it. All LUNs are thin-provisioned. When possible, FlexVol volumes for data and transaction log files of the user database reside on different aggregates.

- FlexVol volume with qtree with LUN for: Data
- FlexVol volume with qtree with LUN for: Transaction log

LUN AND DRIVE LAYOUT

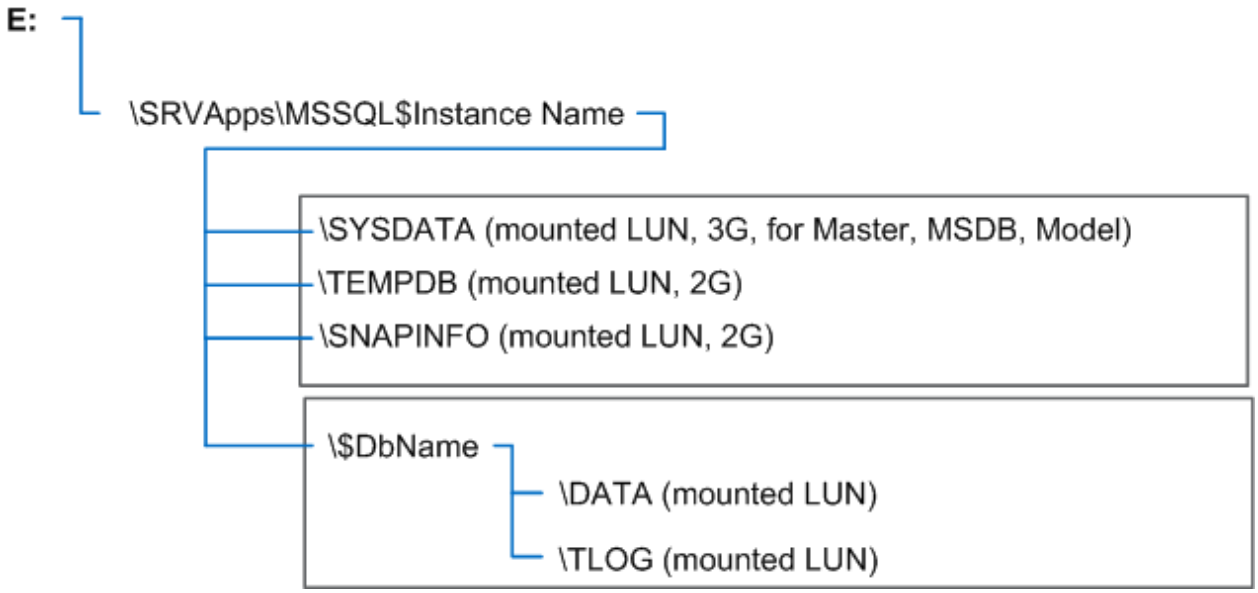
To organize the user databases belonging to a Microsoft SQL Server instance within its folder structure:

1. Create a directory with the name of the user database ($DbName).

2. Mount the LUNs on DATA and TLOG under the created directory structure.

Figure 5 shows the mount points for a small user database.

Figure 5) LUN and drive layout for a small user database.



2.4 LAYOUT FOR LARGE DATABASES (>500GB <1TB SIZE)

This section describes the layout for one or more user databases up to 1TB in size. NetApp recommends that you store the data and transaction logs of a user database using **multiple** LUNs.

This solution utilizes Microsoft SQL Server file group functionality to spread the database and potentially the log files between multiple LUNs.

For each user database, two FlexVol volumes are used. One FlexVol volume is used for data having 10 qtrees, each keeping a LUN. A second FlexVol volume is for the log having a qtree with a LUN in it. All LUNs are thin-provisioned. If possible, the two FlexVol volumes for the data and transaction logs of the user database reside on different aggregates.

- FlexVol volume with 10 qtrees, each with 1 LUN (100GB thin-provisioned) for: DATA
- FlexVol volume with qtree with LUN for: TLOG

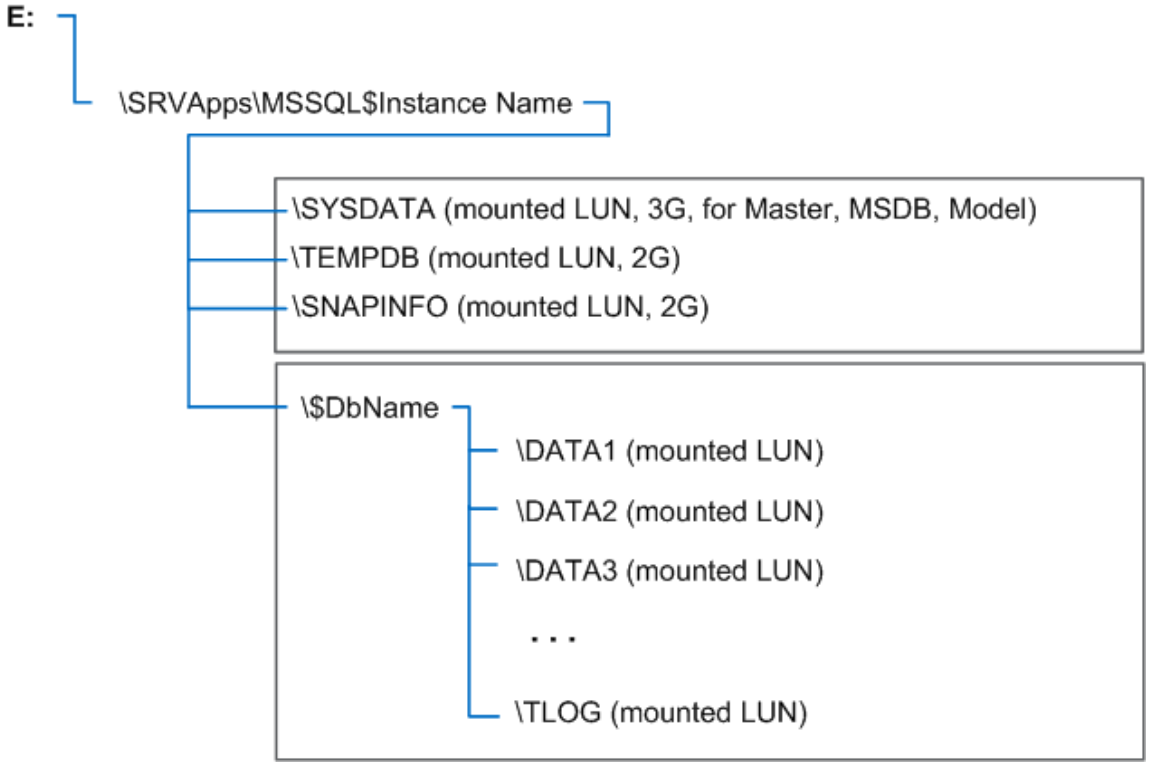
LUN AND DRIVE LAYOUT

To organize the user databases belonging to a Microsoft SQL Server instance within its folder structure:

1. Create a directory with the name of the user database.
2. Mount the LUNs for data1, data2, and so on and for the transaction log under the created directory structure.

Figure 6 shows the mount points for a large user database:

Figure 6) LUN and drive layout for a large user database.



2.5 SUPPORT FOR VERY I/O-INTENSIVE DATABASES

For very large, high-I/O-transaction load databases, the user database files and the transaction log files can be stored on two separate aggregates. Thus the I/O can be off-loaded to two separate groups of spindles. Transaction log files are more sequential and the database is random-access. Separating the two aggregates for large user databases tends to be beneficial and should be taken into consideration.

Note: The platform supports 6,800 to 7,200 IOPS (I/O operations per second) per very large aggregate, which means that most of the databases can safely share the same aggregate for both database and logs. Only in cases of a very high I/O requirement should consideration be given to splitting database and logs between different aggregates.

3 CONFIGURATION RULES

This section covers the individual settings and options that should be used in this configuration.

3.1 WINDOWS CONFIGURATION RULES

The following settings are required to support Microsoft SQL Server:

- Microsoft iSCSI Initiator installed on the Windows host
- Enable multipathing to the NetApp storage array, if required
- NetApp SnapDrive for Windows deployed, configured, and connected to the applicable NetApp storage array

Note: SnapDrive automatically turns off the Snapshot copy creation schedule on the storage system so that the storage system does not create automatic Snapshot copies. All Snapshot copies must be

created either through SnapDrive or through SnapManager for SQL. Currently the maximum LUN size supported by NetApp on Windows is 2TB (2 terabytes).

Note: GUID Partition Tables (GPT) can support a much larger number of blocks, but Windows has some restrictions. Disk devices with more than 2 TB of disk space must be converted to GPT format for all of the disk space to be usable. If the device uses MBR format, the disk space beyond 2 TB will be unusable .SnapDrive supports dynamically increasing the size of a LUN, but LUNs cannot be reverted to their previous size by rolling back to a previous Snapshot copy (taken before the LUN was extended) without data loss.

3.2 MICROSOFT SQL SERVER CONFIGURATION RULES

Customers deploy and support only a limited number of Microsoft SQL Server configurations connected to SnapManager in order to keep the environment as simple as possible and minimize overall support costs.

Note: The Microsoft SQL Server Developer edition is not formally supported by either Microsoft or NetApp since Microsoft does not support it for production purposes. It is supported for development purposes.

Note: The Microsoft SQL Server backward-compatibility module must be installed.

There is no requirement for log shipping between customer sites since this is also handled by SnapMirror replication. Log shipping is not used for making copies of databases for development or reporting purposes. These are produced by using NetApp FlexClone® (writable Snapshot) functionality.

Replication at the Microsoft SQL Server level is an option that the DBA team can explore. It is not documented here, since it does not impact the storage design or layout.

3.3 SNAPMANAGER FOR SQL CONFIGURATION RULES

REPLICATION AND RECOVERY OPTIONS

SnapManager supports multiple replication and recovery options as specified in Table 5.

Table 5) Multiple replication and recovery options.

Operation	Intervals	Mandatory or Optional	Usage
Replication of system databases through SnapMirror to alternate site	Once every 4 hours	Optional	Database Restore
Replication of user databases through SnapMirror to alternate site[2]	Once per hour Once per 6 hours Once per 24 hours	Optional	Database Restore
Stream of database log to Snapinfo directory	Every 15 or 30 minutes	Mandatory	Rapid recovery

[2] Prior to the SnapMirror operation the database is disabled briefly, in order to take the snapshot.

Operation	Intervals	Mandatory or Optional	Usage
SnapVault® backup of user database to NearStore®[3]	Once per day; can be retained for up to 250 days Once per week; can be retained for up to 4.8 years Once per month; can be retained for up to 10 years	Optional	Compliance backups

The options from the previous table are scheduled as detailed in Table 6.

Table 6) Multiple replications with scheduled details.

Operation	Performed By	Scheduling	Default Frequency
Replication of system databases through SnapMirror to alternate site	SnapManager for SQL	Control-M	Every 4 hours
Replication of user databases through SnapMirror to alternate site	SnapManager for SQL	Control-M	Once per hour
Stream of database log to Snapinfo directory	SnapManager for SQL	Control-M	Once per 15 minutes
SnapVault backup of database to NearStore	SnapManager for SQL	Control-M	Each week night
Physical verification of database files	SnapManager for SQL	Control-M	Each week night
Reindexing of tables	Microsoft SQL Server	Control-M	Monthly

Table 7 provides the authority for determining the replication and recovery options.

Table 7) Replication operation with scheduling details.

Operation	Scheduling
Replication of system databases through SnapMirror to alternate site	Database administrator
Replication of user databases through SnapMirror to alternate site	Business owner
Stream of database log to Snapinfo directory	Database administrator
SnapVault backup of database to NearStore	Business owner

[3] There is a limit of 250 Snapshot copies, which limits the maximum retention period.

SNAPMANAGER FOR SQL CONFIGURATION LIMITS

Table 8 represents SnapManager for SQL configuration limits for the currently released version.

Table 8) SnapManager for SQL configuration limits.

Configuration Capacity	Maximum Supported with SMSQL 5.1
SQL Server instances per SQL Server computer or Windows cluster	Windows Cluster – 25 Standalone Windows host for SQL Server 2005 and 2008 – 50 SQL Server 2000 – 16
LUNs per SQL Server computer	115
VMDKs per SQL Server computer	56
Databases per LUN or VMDK	100
Databases per storage system volume	100
Databases per SQL Server instance	Virtual instance – 1,000 Standalone instances – 2,500
File groups per database	75
Storage system volumes that can be used to store the following:	A single database – 30 LUNs connected to an individual SQL Server – 120 VMDKc connected to an individual SQL Server – 56
Max NFS datastores	64
Fibre Channel LUNs/host	256
Paths to LUN	32
Software iSCSI initiators	
LUNS/Host	2,563
LUNS concurrently connected	256
Total paths to LUN	1,024

Note: Each instance need not be configured or defined to the Microsoft SQL Server. Only the first instance must be defined on the initial build.

Note: The table entries are directories, not mount points.

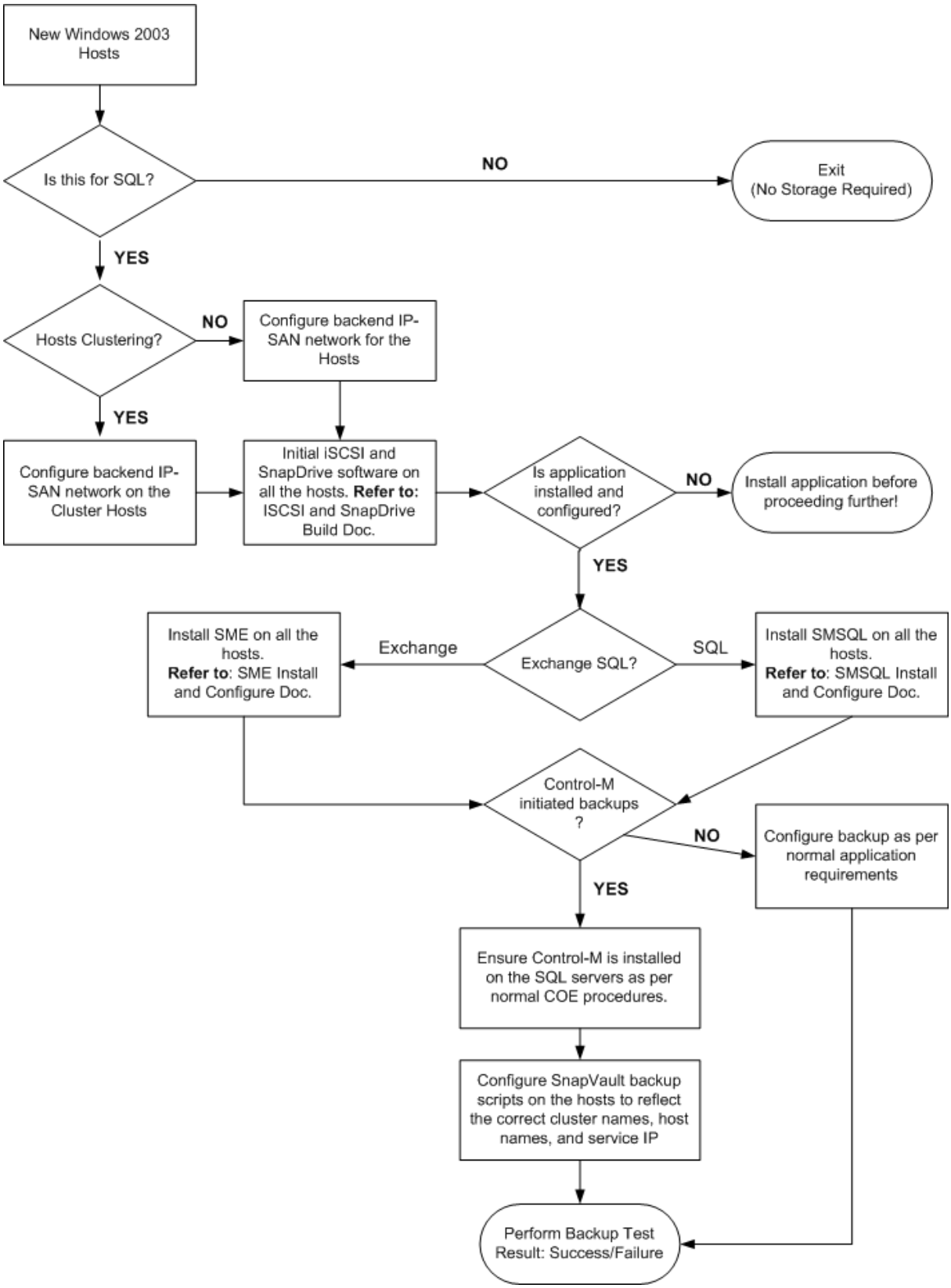
Note: You must edit the default NFS datastores value if you plan to add more than the 8 default datastores. The maximum value is 64.

This task of relocating the database/log must always be accomplished by using the SnapManager for SQL configuration wizard.

HOST SOFTWARE INSTALLATION PROCESS FOR SNAPMANAGER FOR SQL

Figure 7 shows the process flow for the installation of host software applications for SnapManager for SQL and their dependencies.

Figure 7) Host backup software installation.

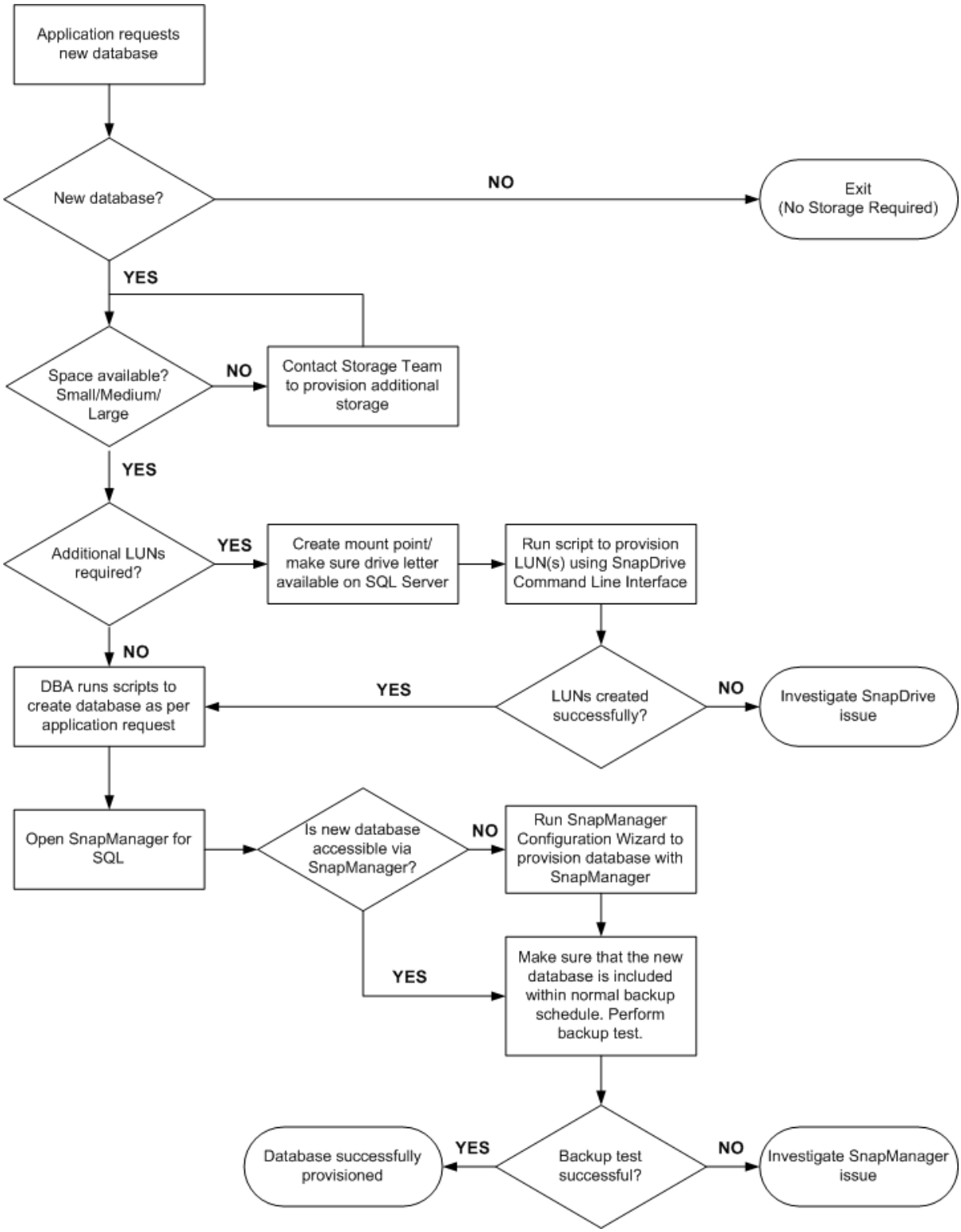


SMSQL—BACKUP PROCESS FOR A NEW MICROSOFT SQL SERVER DATABASE

Figure 8 shows the process flow for new database creation and the steps required for SMSQL to back up the databases. This process must be followed whenever a new database is created on the Microsoft SQL Server instance.

Make sure that the SMSQL configuration wizard is running before adding, removing, or moving databases to different or new LUNs. If it is not, you could create invalid configuration information within SMSQL.

Figure 8) SMSQL backup process.



3.4 SNAPMANAGER FOR SQL—ADDITIONAL INFORMATION

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

- Cloning a database from an existing backup set
- Cloning active production databases
- Cloning a database on a SnapMirror destination volume
- Deleting a cloned database

This is of great value in a test/dev environment in which real-time access to OLTP databases is required. If a database resides on a virtual machine with a VMDK disk, it is not possible to clone the database to a physical server.

Figure 9) Different cloning options.

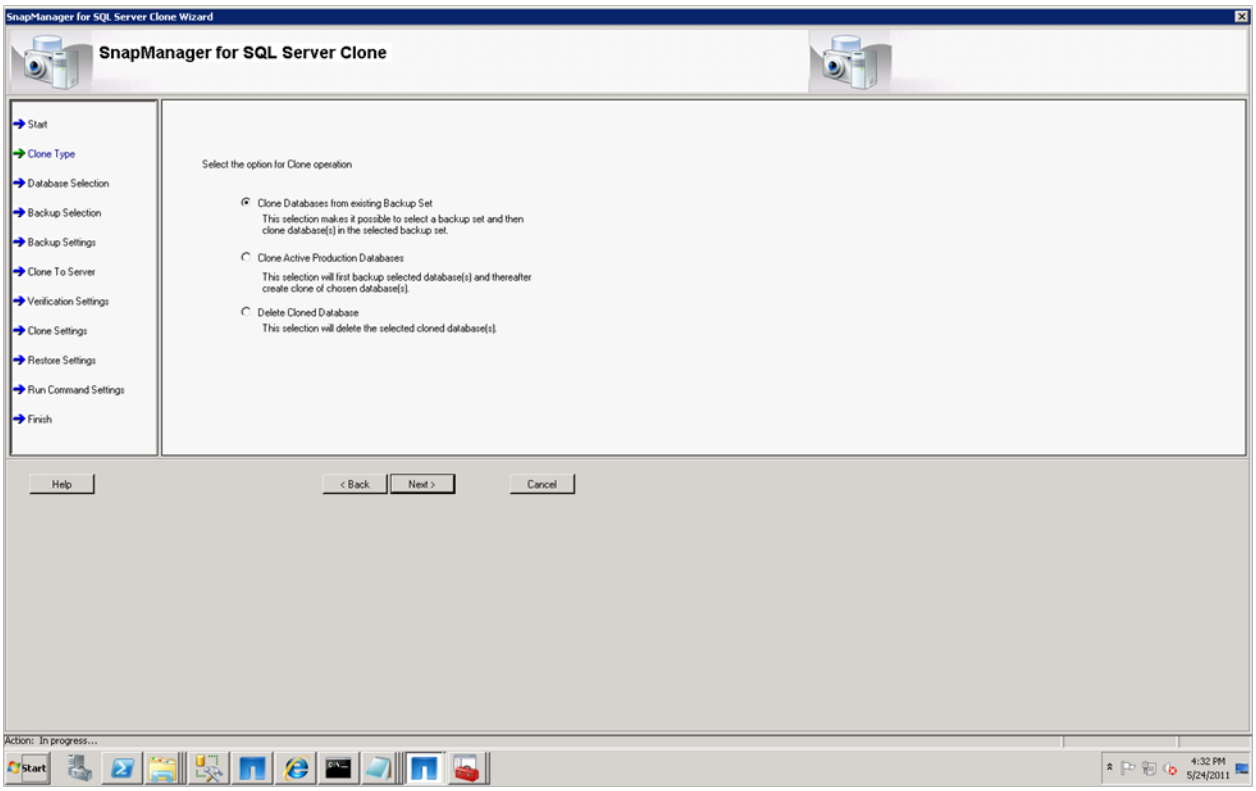


Figure 10) Databases to be cloned.

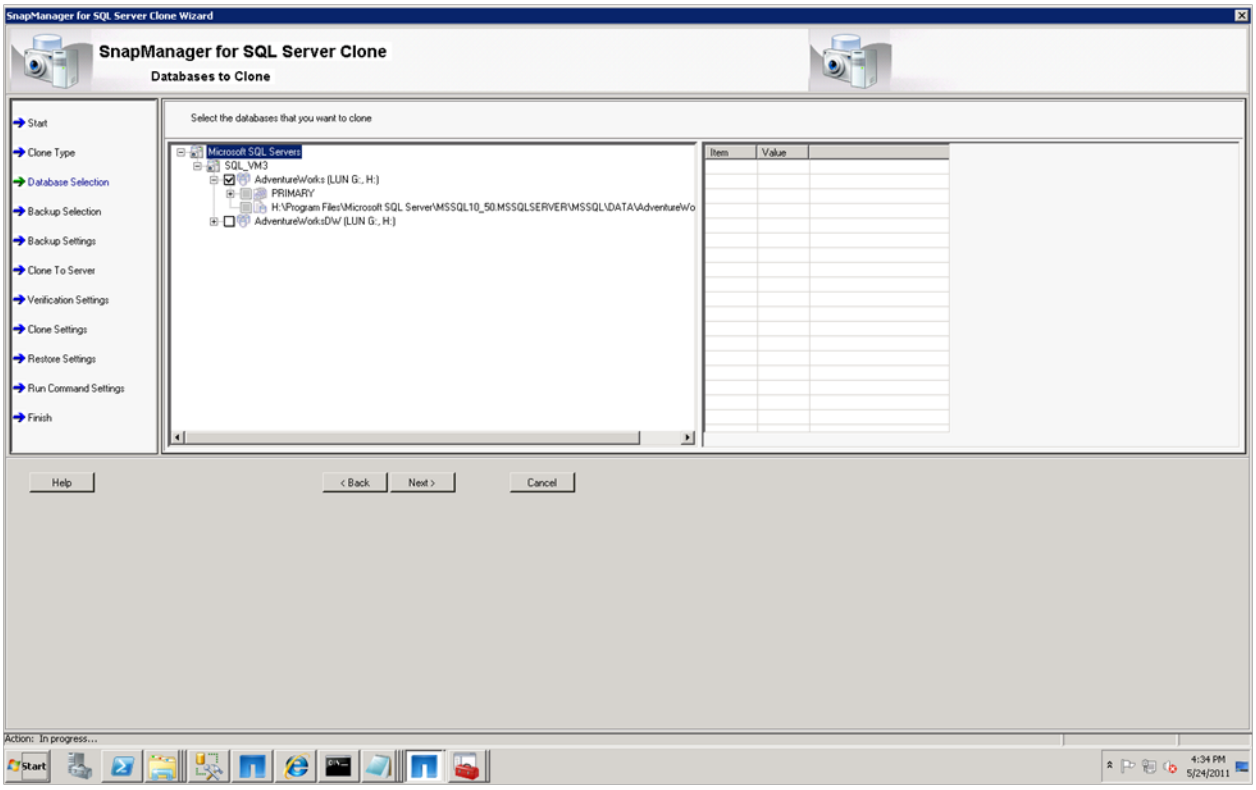


Figure 11) Naming the cloned database.

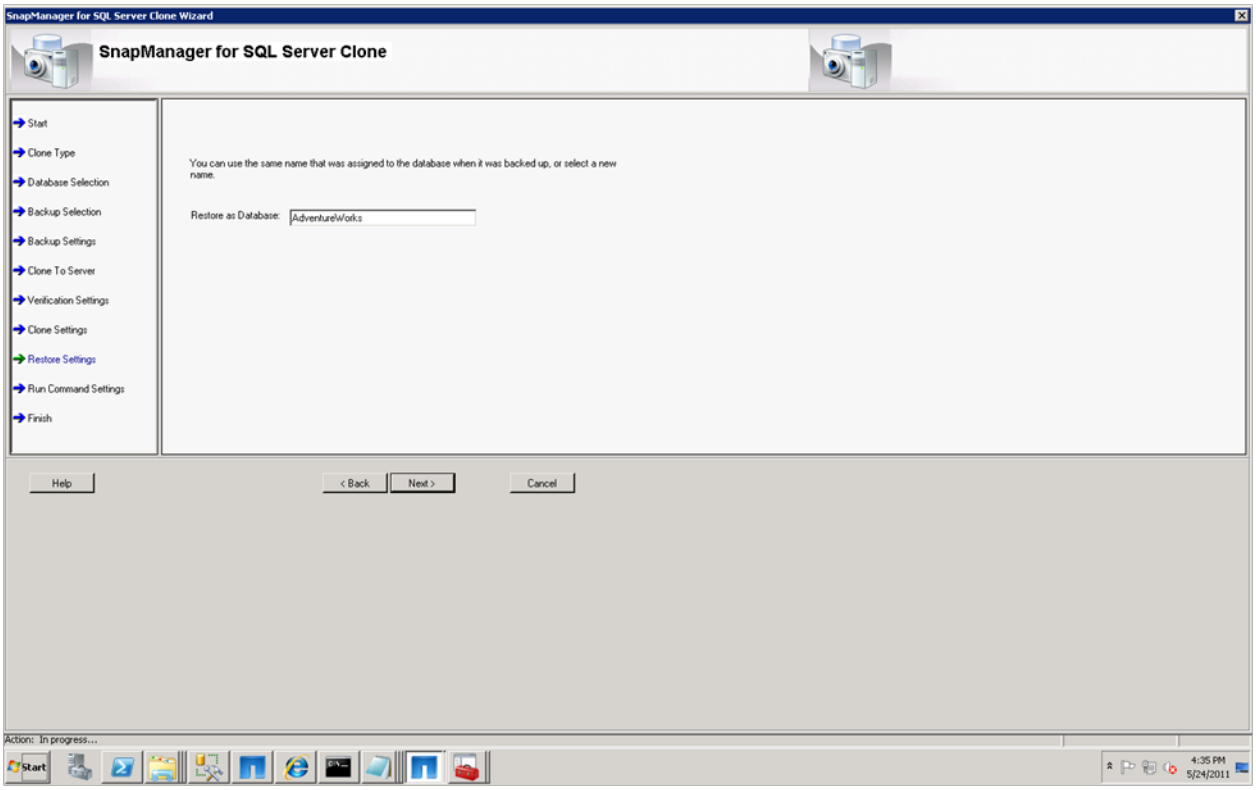


Figure 12) Clone settings.

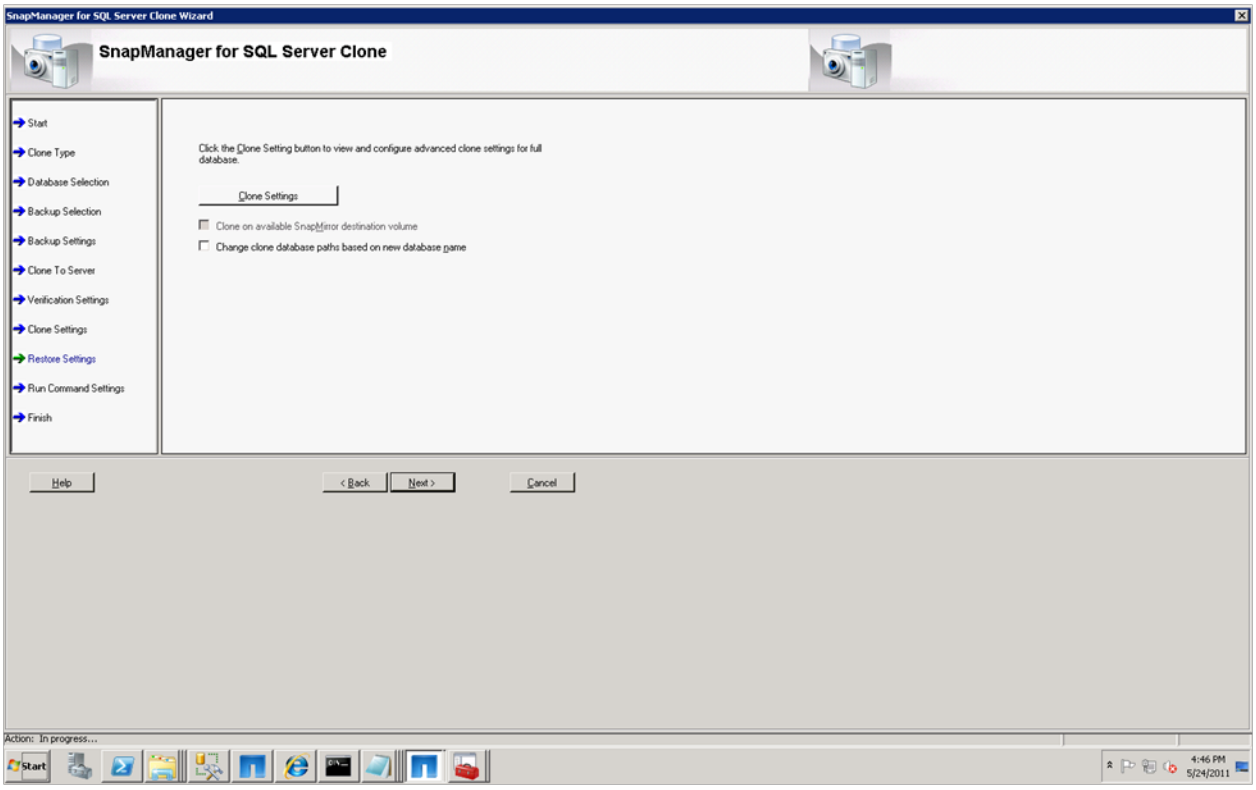
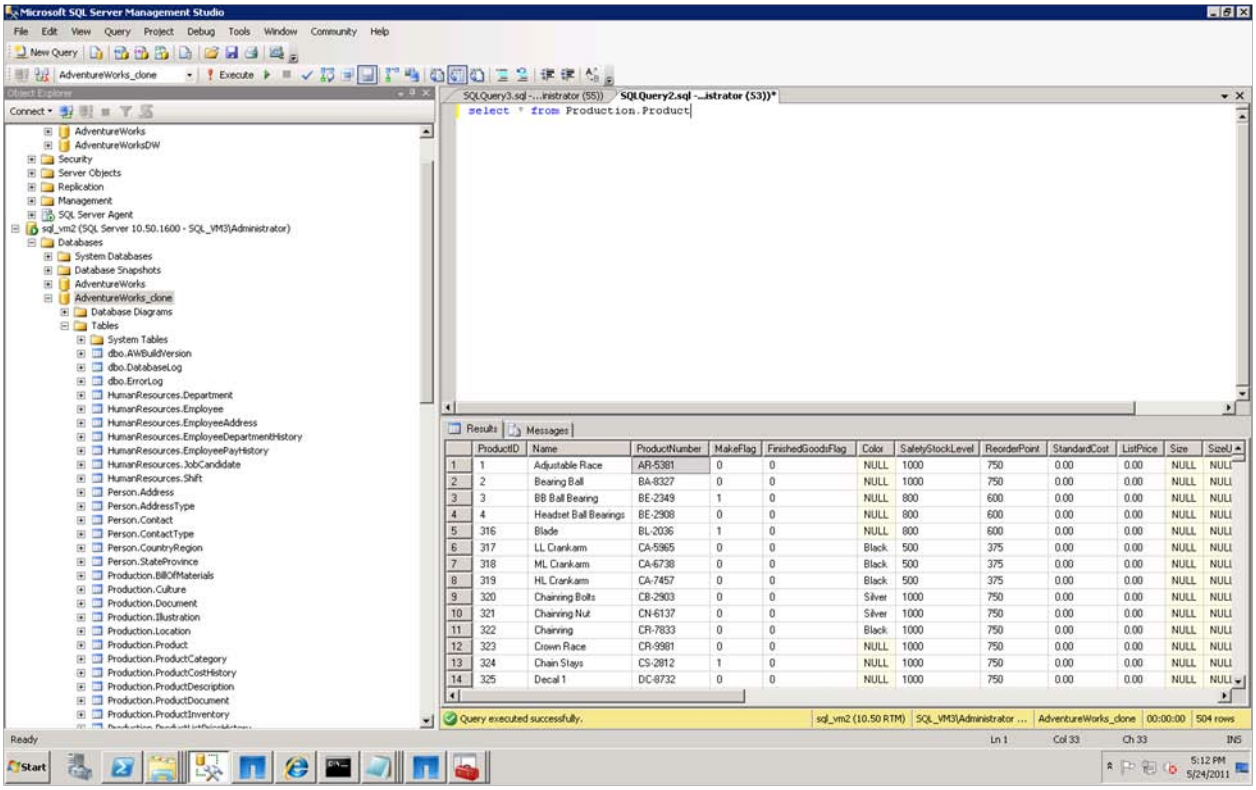


Figure 13) Querying the cloned database.



MOUNT POINT NAMING CONVENTION

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

```
VDISK-MountPoint-n
```

where n represents the configuration number of a mount point in a specific system. The actual mount point paths are saved in the Windows registry.

LUNS WITH 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 the mount point root. For example:

```
LUN E: (MPRoot)
```

This SMSQL configuration wizard does not allow storing Microsoft SQL Server database files on LUNs that host NTFS volume mount points.

SNAPMANAGER SERVICE ACCOUNT

To run the SnapManager backup and restore functions, the SnapManager service account must have 'sysadmin' fixed server role privileges on all Microsoft SQL Server instances on the physical machines. This is usually addressed by giving the Windows user account administrator rights on the Microsoft SQL Server.

This can be performed by making the SnapManager for SQL service account a member of one of the following groups:

Table 9) AD groups with different environments.

Environment	AD Group
Production	PROD\PRD-Role-Svc-SQL-SnapManager
Development	DEV\DEV-Role-Svc-SQL-SnapManager
Discovery	TEST\DIS-Role-Svc-SQL-SnapManager

CONCURRENT BACKUP OF MULTIPLE DATABASES

When SMSQL backs up one or more databases managed by a Microsoft SQL Server, the Microsoft SQL Server requires internal resources (worker threads) to manage each database that is being backed up. If insufficient resources are available, the backup operation fails. The SnapManager configuration wizard provides the number of databases that can be backed up concurrently.

Note: Concurrent backup of multiple databases is considered unlikely and this parameter must be changed.

If the number of databases being backed up is more than 40, consider modifying the maximum worker threads parameter. The value can start from 128 increments.

3.5 RUN THE MICROSOFT SQL SERVER CONFIGURATION WIZARD

The SMSQL configuration wizard must be run in the following situations:

- For initial configuration: To use SMSQL to back up and restore the Microsoft SQL Server databases, SMSQL configuration must be run to migrate the databases to NetApp LUNs and also to set up the Snapinfo directory for use by SMSQL.
- To view or change the database configuration: The configuration wizard can be run to view or make changes to the Microsoft SQL Server database configuration.

- To validate the database configuration: While adding more databases or moving databases to different LUNs without using SnapManager, the Microsoft SQL Server configuration wizard must be run to avoid any configuration errors.
- When a new database is added to allow backups.

Note: Currently the Microsoft SQL Server configuration wizard must be run interactively. Later versions may support scripting, so these steps could be automated.

3.6 CONFIGURATION WIZARD

Note: The current release only allows the configuration wizard to run interactively. Later versions may support automation through either the command line or through a parameter file.

The configuration wizard enables the user to move Microsoft SQL Server data in the following ways:

If the database is going to be moved from local disk to LUN to enable management by SMSQL:

- Microsoft SQL Server databases must be moved. The wizard dismounts the databases in a storage group, moves the Microsoft SQL Server database and transaction log files to the selected LUN, and remounts the databases.

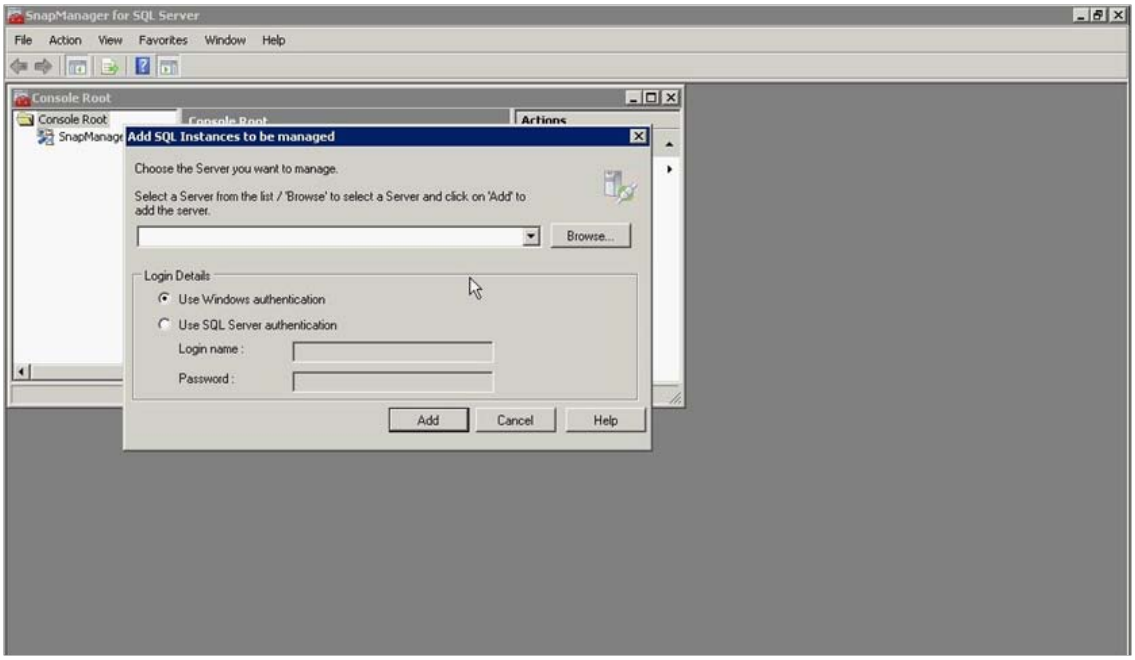
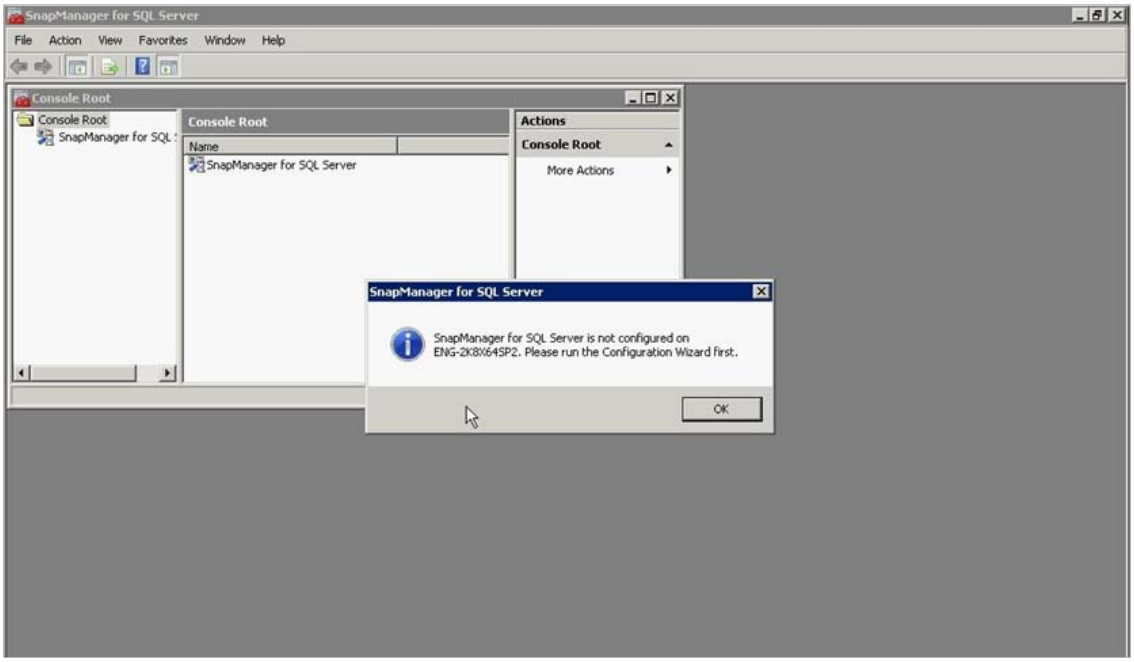
Note: SnapManager takes Microsoft SQL Server databases offline during the move operation.

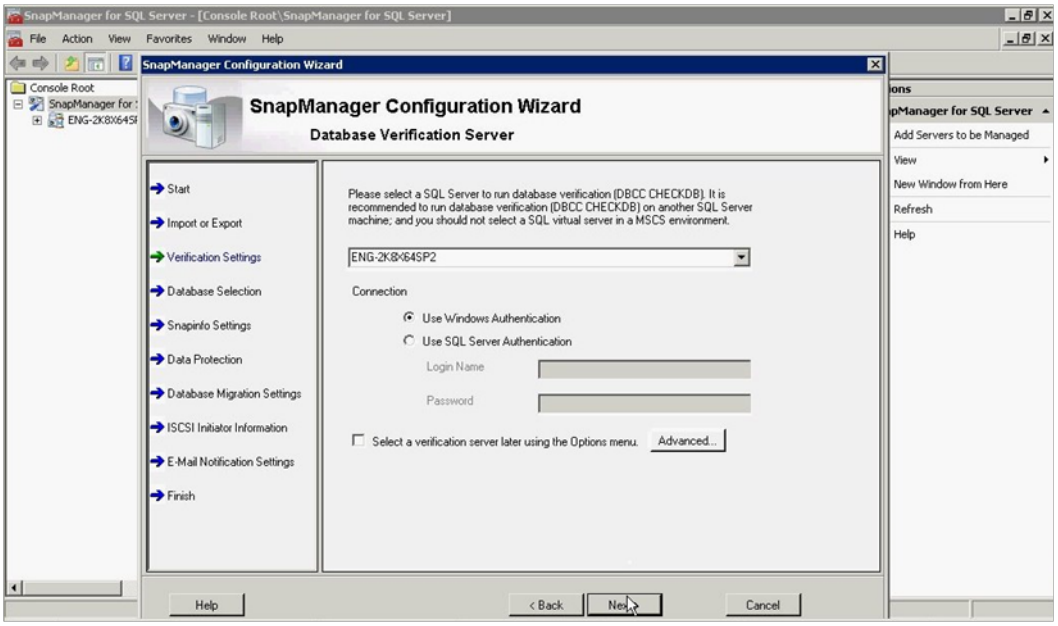
- The wizard creates a Snapinfo directory that SnapManager uses to store information about the backup sets and the backed-up transaction logs.
- The wizard goes through several application settings. These settings include enabling notification of SnapManager events using either e-mail or the storage system Syslog or the AutoSupport™ feature.

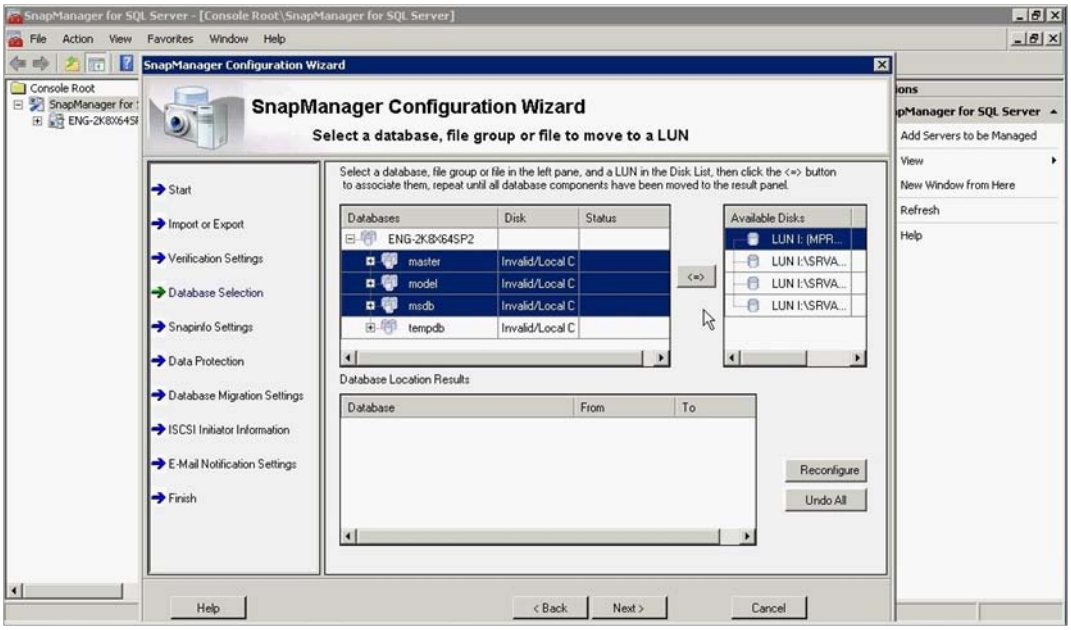
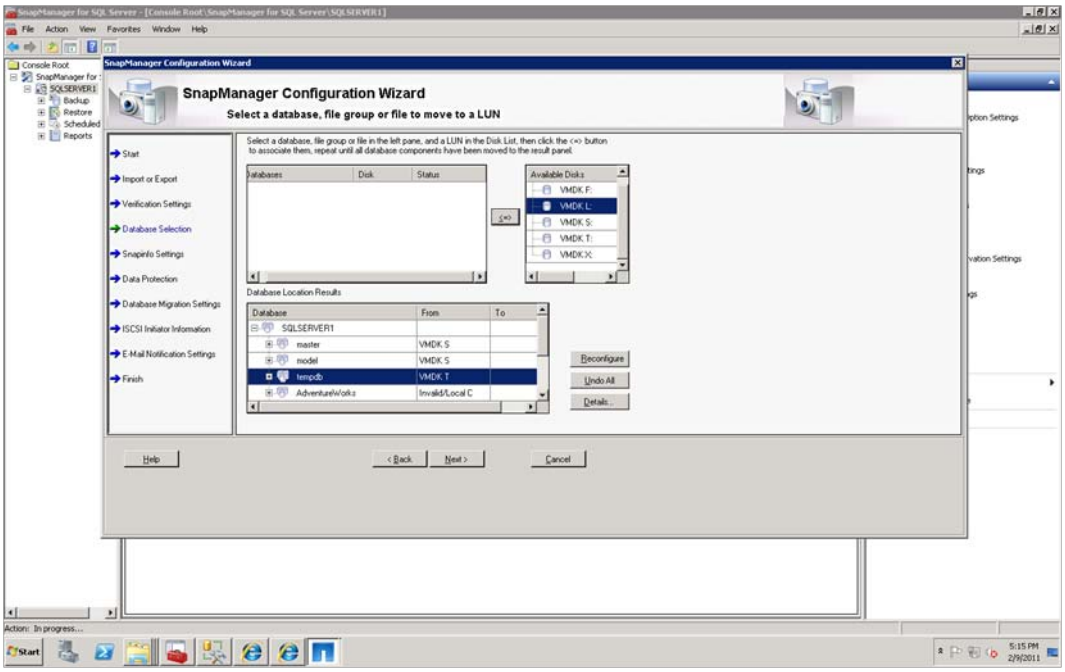
Note: If the Microsoft SQL Server database and logs are already on the LUNs but the SMSQL configuration wizard was never run, the SMSQL configuration wizard automatically determines the location of the database on the LUNs upon running. The SMSQL configuration wizard must still be run so that SMSQL is configured and the Snapinfo directory is created for the backup sets.

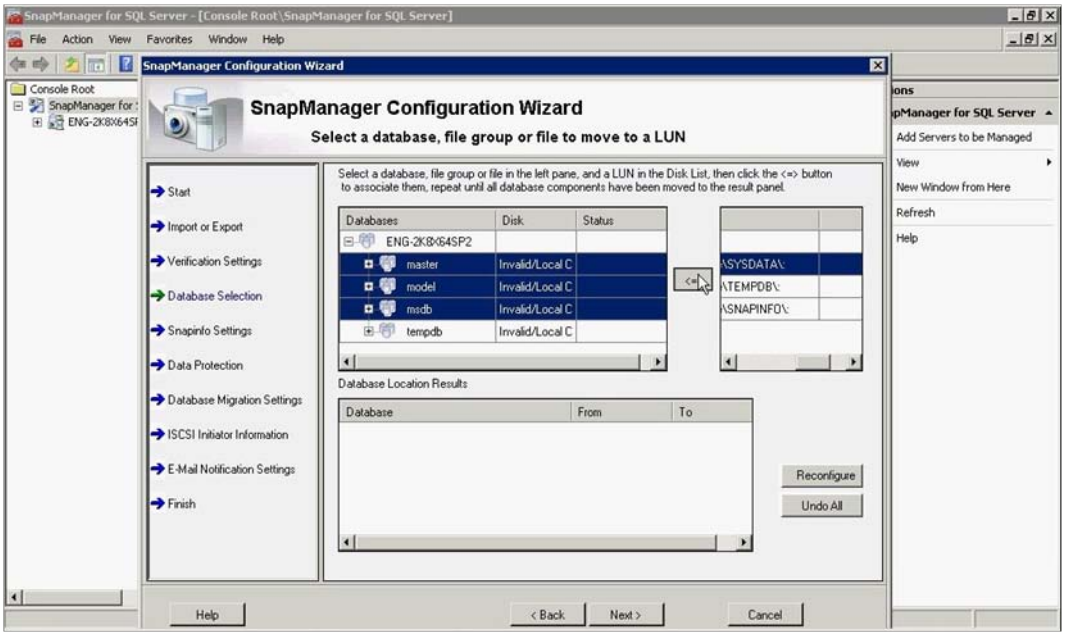
The procedures outlined in Table 10 detail the configuration of SnapManager for SQL Server.

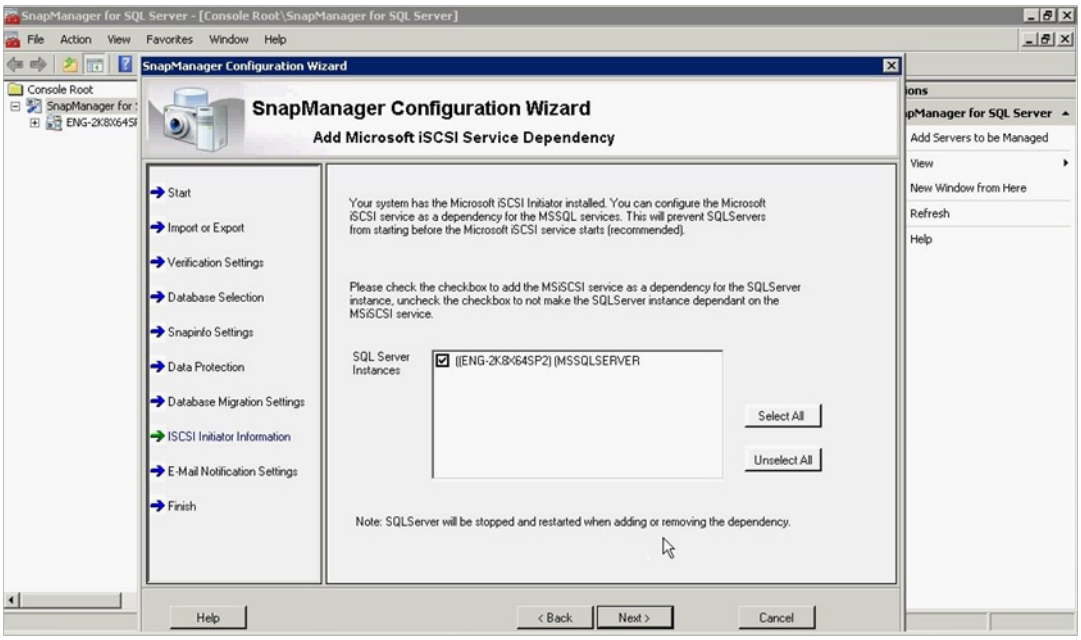
Table 10) SnapManager for SQL Server configuration wizard.

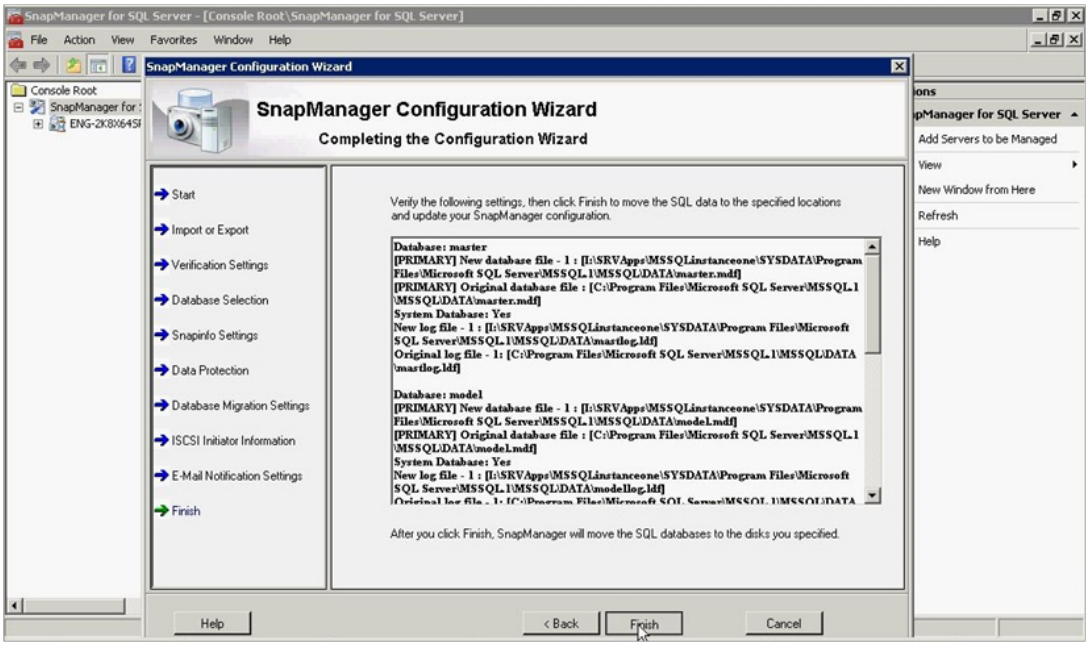
Step	Usage
1.	Start SnapManager. If this is the first time SnapManager has started, the dialog to connect to SQL Server is displayed. Specify the correct SQL Server and the connection authentication type here: 
2.	A notification informs you that the SQL Server instance is not configured for SnapManager. 
3.	Click OK. Click OK again. If this is the first time that SnapManager has been started after installation, a dialog box pops up stating that SnapManager for SQL Server is not configured on the server. Click OK and then click Next.

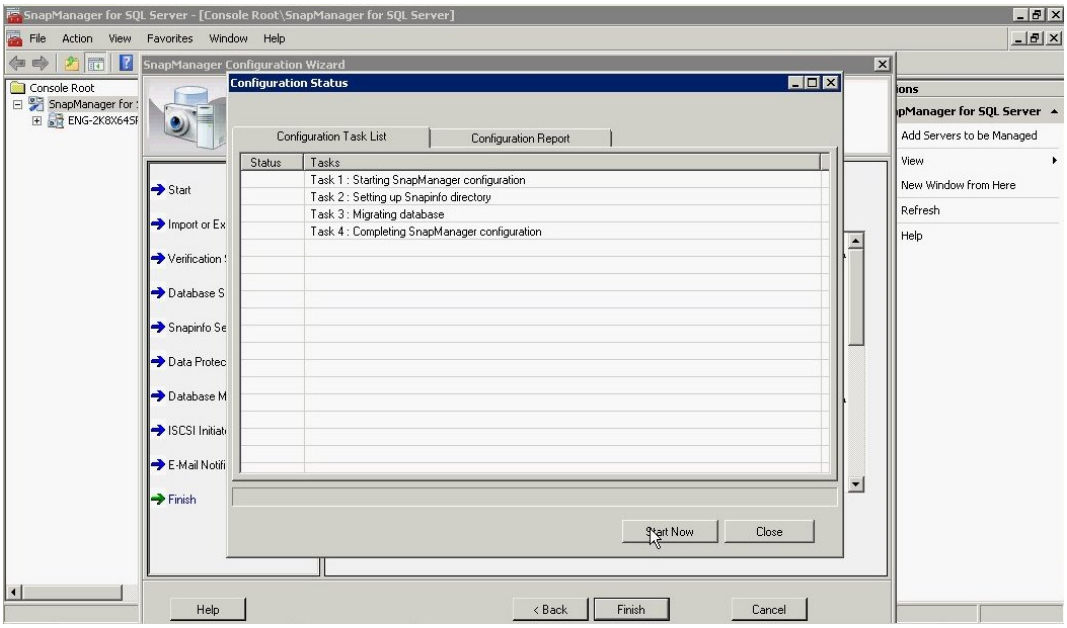
Step	Usage
4.	Select the verification server and determine that the appropriate connection authentication is selected. Click Next. 

Step	Usage
5.	Specify the database in the window that is displayed.  If you are using a VMDK LUN you will see this screen: 

Step	Usage
6.	Pair databases, file groups, or files in the left pane with the LUN from the right pane according to the storage layout. When you are finished, click Next. 
7.	Select a single Snapinfo directory for a given SQL Server and click Next. Select the current Snapinfo directory and then highlight the disk in the right-hand pane. Click the arrow button and then click Next.  The result is the new Snapinfo directory that stores all the backup sets. **Note:** Multiple databases always share the same Snapinfo directory.

Step	Usage
8.	Select the default data migration options. Click Next. 
9.	Select the SQL Server instances and add Microsoft iSCSI service dependency. Click Next. 

Step	Usage
10.	Configure the automatic event notification and click Next. 
11.	Complete the configuration wizard by verifying the settings selected for SMSQL. Click Finish. 

Step	Usage
12.	Click the Start Now button to see the configuration status. 
13.	On successful completion, SMSQL notifies the user.

After completing the preceding steps, the actual SMSQL window shows the Microsoft SQL Server in SnapManager (Figure 14) and can be used to take Microsoft SQL Server backups and to perform restores.

Figure 14) Microsoft SQL Server in SnapManager.

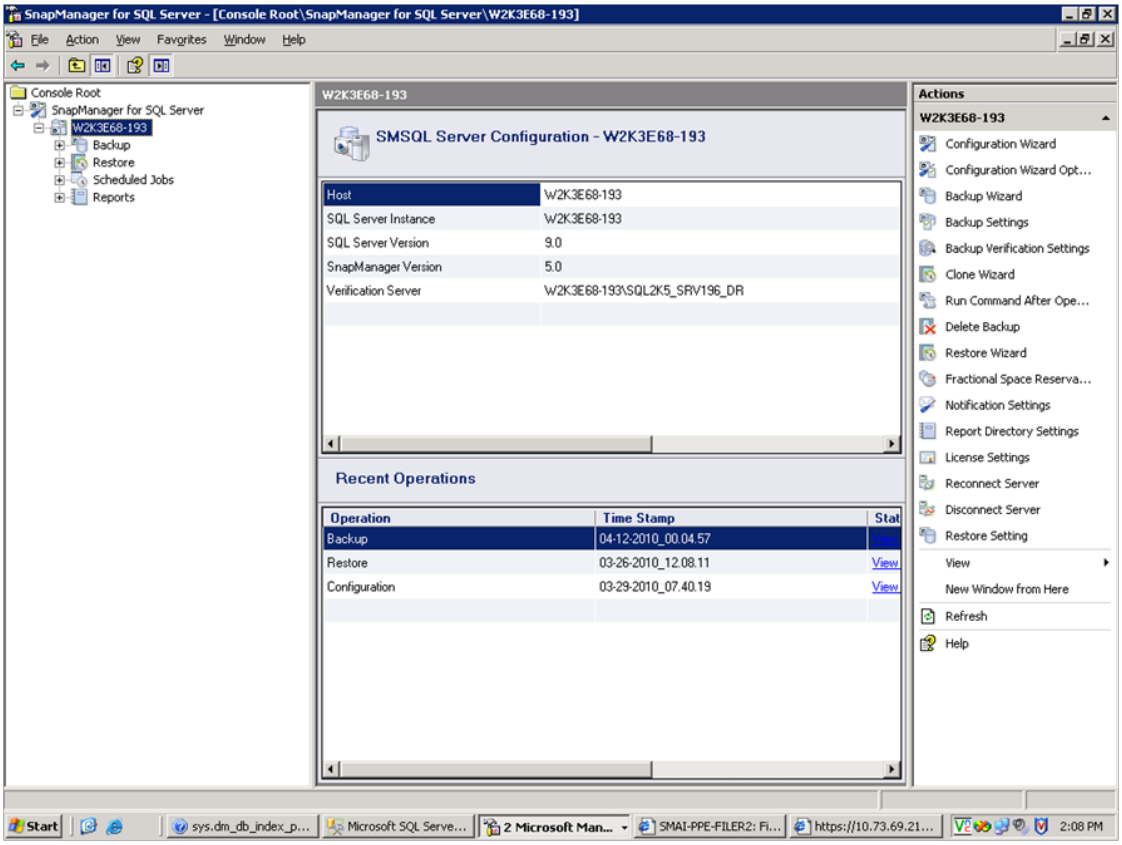
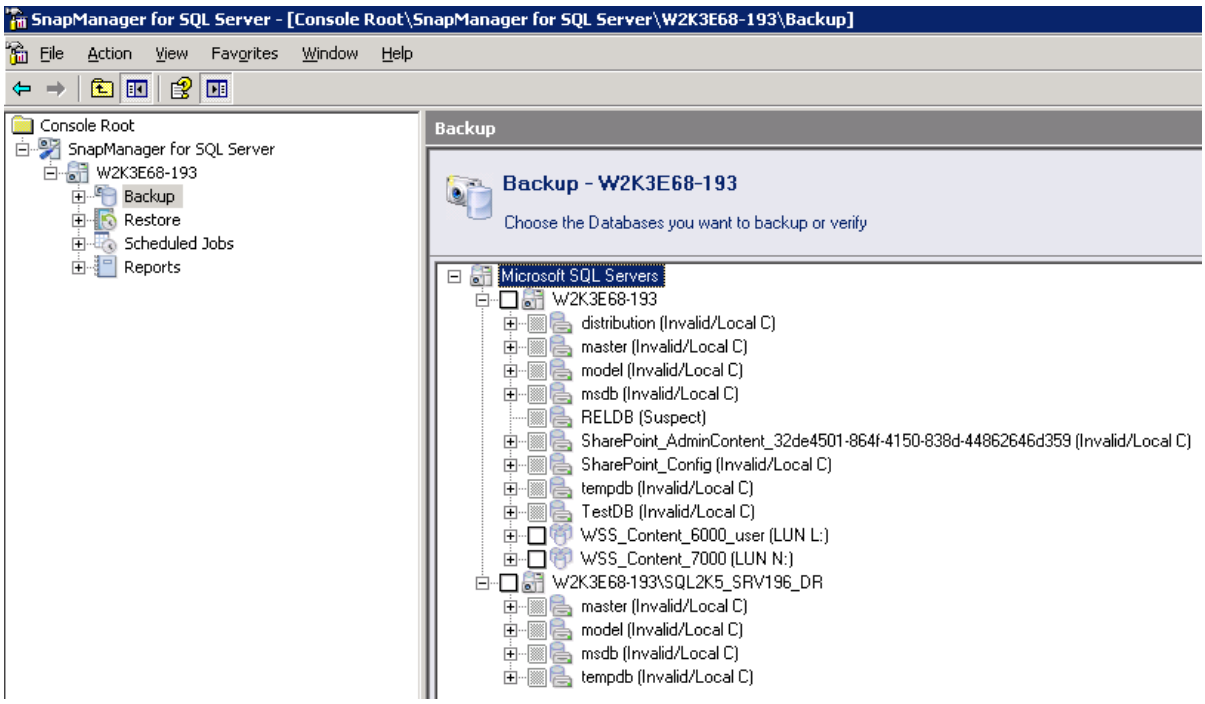


Figure 15) Microsoft SQL Servers.



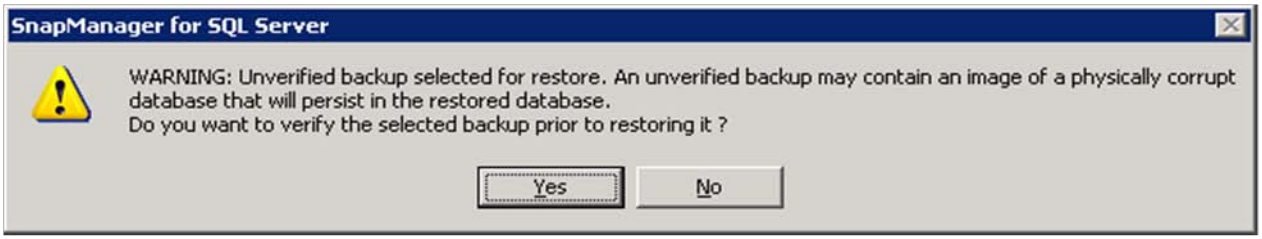
4 DATABASE VERIFICATION

4.1 DATABASE CONSISTENCY CHECK—DATABASE VERIFICATIONS USING SMSQL

As part of the SMSQL backup of a Microsoft SQL Server database, SnapManager performs a Database Consistency Check (DBCC) against a database. By default, the DBCC check is run directly after the backup is completed. This job is run during off-peak hours to prevent excessive load on the production system. Generally the DBCC check is run against the databases on a nightly basis. Customers can also do the verification settings (DBCC CHECKDB) on a remote server that may not be the OLTP production DB.

By default SMSQL does not let a DBA restore a backed-up database that is not checked for inconsistencies, but it provides the option to let a database be restored, even when consistency is not verified. The DBA has to acknowledge, during the restore process, that the safety net is bypassed, as shown in Figure 16.

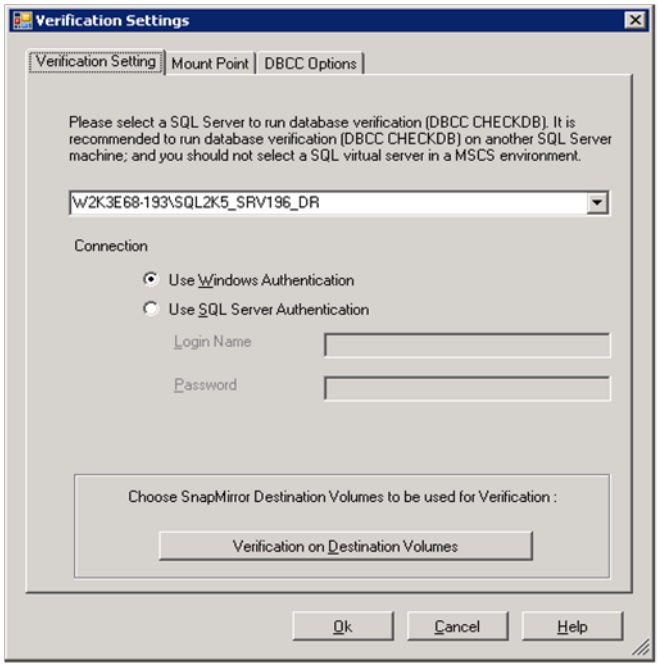
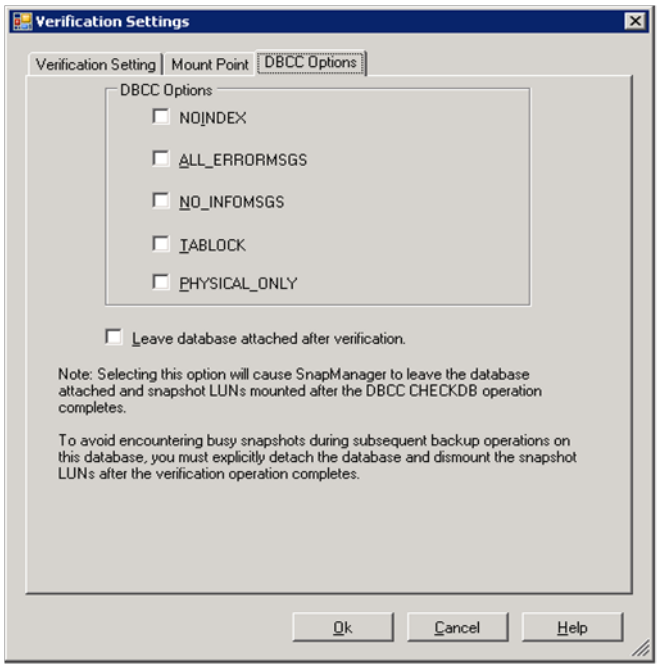
Figure 16) SnapManager for SQL Server.

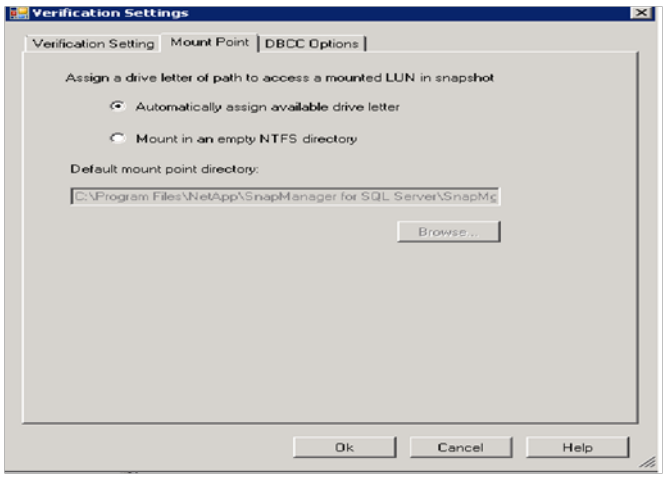


Note: Verification of backed-up databases must be performed by a nonclustered Microsoft SQL Server instance either installed on the local Windows server or on another Windows server that can be a dedicated verification server. This is because a virtual Microsoft SQL Server cannot be used for verifying a database's consistency. The LUN in the Snapshot copy is restored during the verification process, and the temporary restored LUN cannot be a cluster resource and therefore cannot be included in Microsoft SQL Server Server's dependency list. A virtual Microsoft SQL Server instance can be used for verifying the online database.

Perform the steps in Table 11 to verify the backed-up databases.

Table 11) Database verification.

Step	Action
1.	To specify the verification server, go to SMSQL → Options menu → Verification Settings and specify or select the SQL Server to run the verification. 
2.	Click the DBCC Options tab in the Verification Settings window and select the options required. NetApp recommends leaving it as the default. 

Step	Action
3.	Click the Mount Point tab and specify whether to use a mount point or drive letter as part of the verification. 

FILE-LEVEL BACKUP MOUNT POINT

This drive is mounted into a directory named `\SRVApps\MSSQL$InstanceName\Backup`. It contains the file-based backups of Microsoft SQL Server that are required. This drive is backed up once a day. The size of this drive varies depending on the specific instance. Its minimum size is 1GB.

Each instance of Microsoft SQL Server requires the following directory flow depending on the LUNs per database.

Table 12) Drive size with path.

Path	Minimum Size	Description
D:\	3GB	Stores the directory structure for all the mount points
D:\SRVApps\MSSQL$InstanceName\Data\Datan	1GB	Stores the database and file group files: n represents directory for each mounted LUN per database
D:\SRVApps\MSSQL$InstanceName\Data\Tlogn	1GB	Stores the active transaction log files: n represents directory for each mounted LUN per database
D:\SRVApps\MSSQL$InstanceName\Data\TempDB	2GB	Stores the database's temporary database
D:\SRVApps\MSSQL$InstanceName\Data\SysData	512MB	Contains the MASTER, MODEL, and MSDB databases

Here D: is replaced with the appropriate drive letter for the server/cluster/instance being configured. InstanceName is replaced with the name of the instance to be installed.

5 CONFIGURATION OF CONTROL-M SCHEDULING

Control-M scheduling can be set up as detailed in Table 13.

Table 13) Control-M scheduling.

Operation	Control-M Job Name	Command Line
Replication of system databases through SnapMirror to alternate site	SMSQL-SM-SYSTEM_$DBname	SMSQL…
Replication of user databases through SnapMirror to alternate site	SMSQL-SM-USERDB_$DBname	SMSQL…
Stream of database log to Snapinfo directory	SMSQL-LOG_$DBname	SMSQL…
SnapVault action of database to NearStore	SMSQL-SV-USERDB_$DBname	SMSQL…
Physical verification of database files	SMSQL-VERIFIY_$DBname	SMSQL…
Reindexing of tables	SQL-REINDEX_$DBname	SQL Maintenance Wizard

6 APPENDIXES

6.1 DEFINITIONS

This section contains a glossary of terms used throughout this document.

Table 14) Glossary of terms.

Term	Definition
CIFS	Common Internet File Service
DNS	Domain Name System
DR	Disaster Recovery
DRC	Disaster Recovery Center (Data Center)
FAS	Fabric Attached Storage
FC	Fibre Channel
FlexVol	Flexible Volume
IOPS	Input/Output Operations per Second
iSCSI	Internet Protocol Small Computer Systems Interface
MAN	Managed / Metro Area Network
LUN	Logical Unit Number
NAS	Network Attached Storage
NFS	Network File System
NIS	Network Information Service
NTP	Network Time Protocol

Term	Definition
PDU	Power Distribution Unit
PDC	Primary Data Center
RPM	Rotations per Minute
RAID	Redundant Array of Independent Disks
SAN	Storage Area Network
SNMP	Simple Network Management Protocol
SATA	Serial Advanced Technology Attachment
UPS	Uninterruptible Power Supply
VIF	Virtual Interface
VLAN	Virtual Local Area Network
VMDK	Virtual Machine Disk
WINS	Windows Internet Naming Service

6.2 PREREQUISITES FOR INSTALLING SNAPMANAGER FOR SQL

The exact prerequisites are documented in the SnapManager release notes.

The following combinations are supported, but the release notes contain the definitive list:

- Microsoft SQL Server 2008 on Windows 2008
- Microsoft SQL Server 2008 R2 on Windows Server 2008 R2

The various editions of Microsoft SQL Server are supported (Standard and Enterprise), while the Developer edition is not formally supported because Microsoft does not support it for production purposes. It is supported for development purposes only.

SMSQL is packaged as an application and deployed as per the normal procedures.

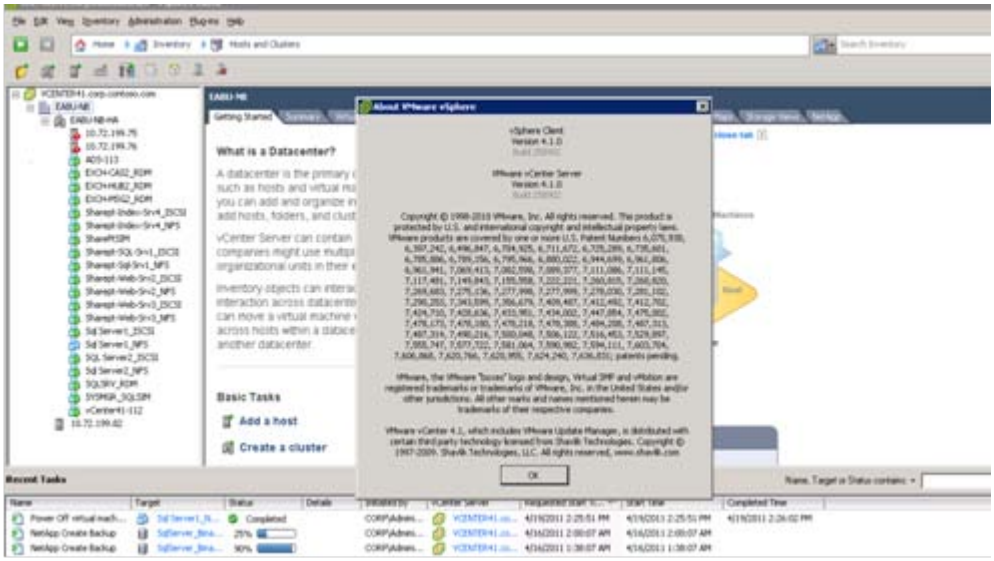
The following requirements must be met before installing SMSQL on the Windows servers running Microsoft SQL Server 2008:

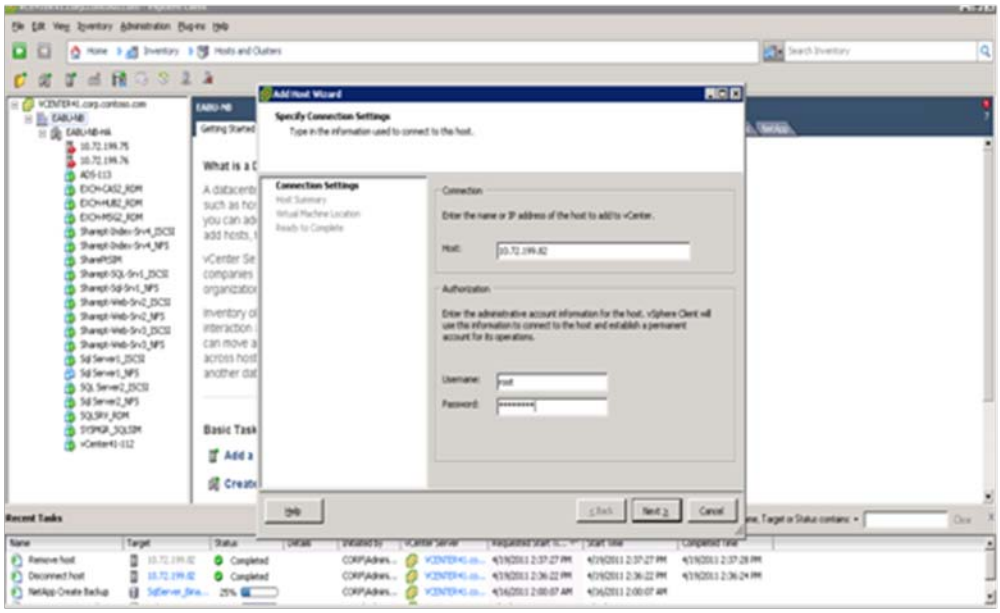
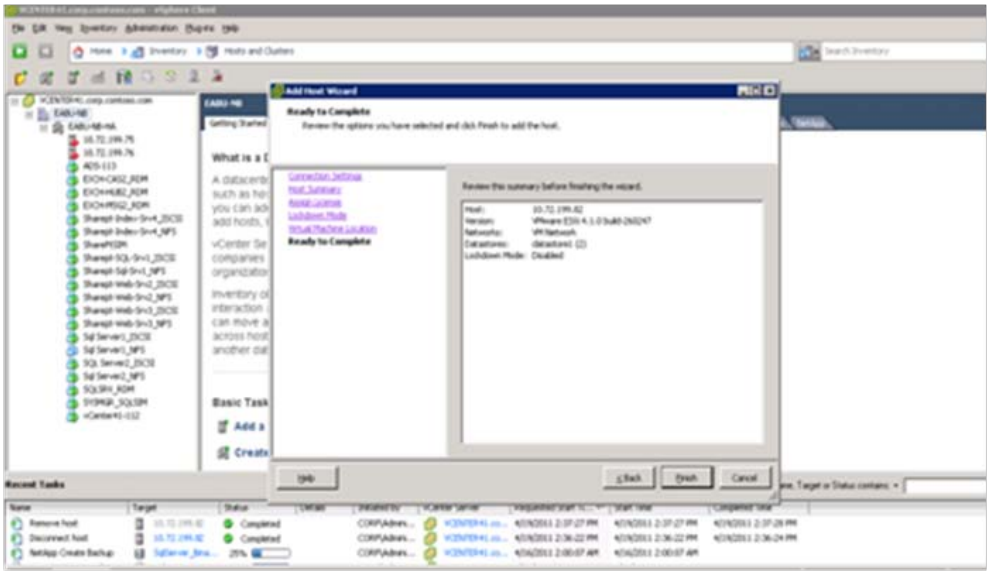
- Make sure that Microsoft Direct Access Components (MDAC) 2.8 SP2 is installed.
- If the host system is running Microsoft SQL Server 2008, configure the Microsoft SQL Server browser service to start automatically.
- The iSCSI S/W initiator and SnapDrive for Windows must be installed.
- Storage must be configured and the LUNs must be presented to the Windows server(s).
- SMSQL runs as a service on Windows servers and requires a service account. The SMSQL service account must be the same on all servers in an MSCS cluster.
- To set up the right level of permissions for the SnapManager for SQL account, read the SnapManager Service Account section in this document.
- If using NFS for configuration, please configure Virtual Storage Console (VSC) on VCenter™.
- Create an NFS/iSCSI/FC datastore from vCenter.

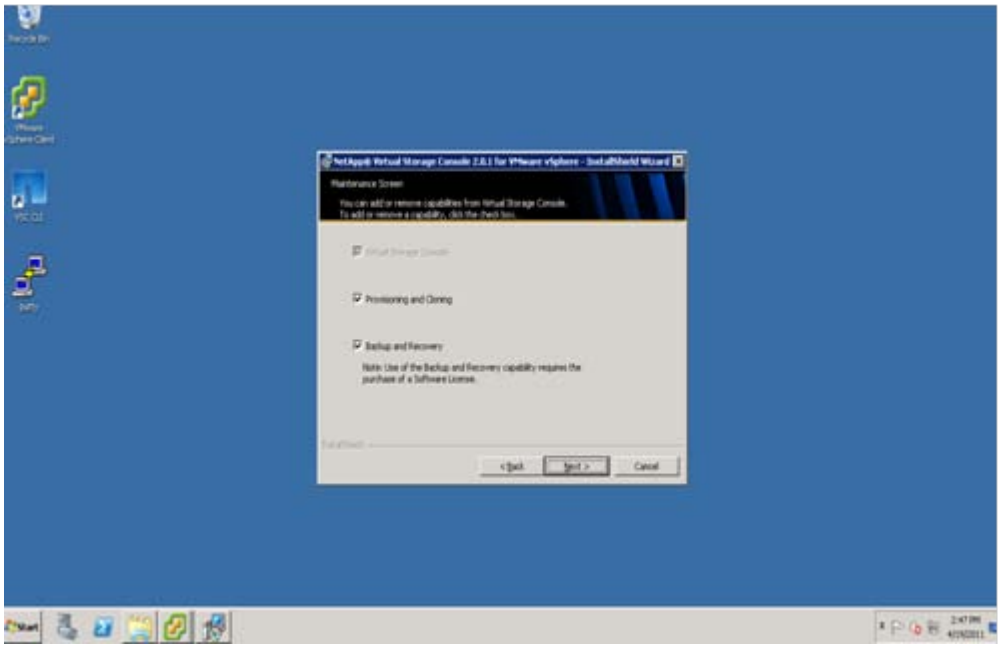
PREREQUISITES FOR CONFIGURING SNAPMANAGER FOR SQL SERVER ON VMDK

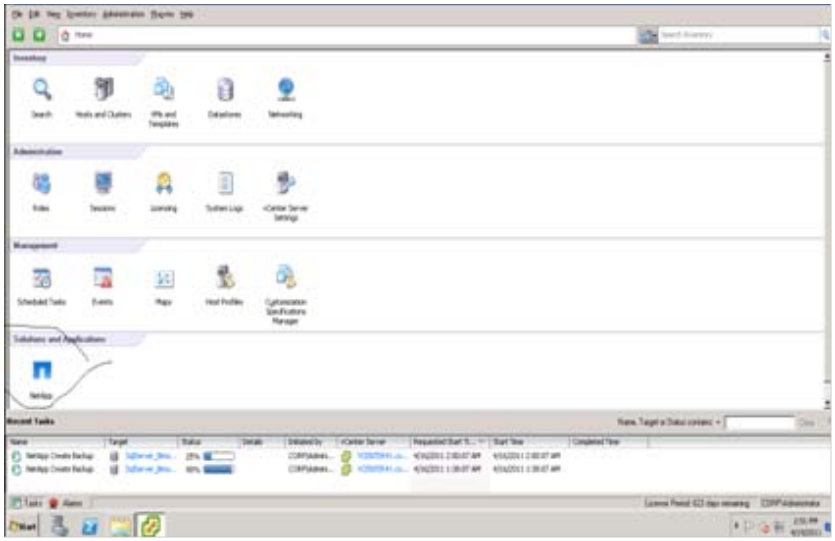
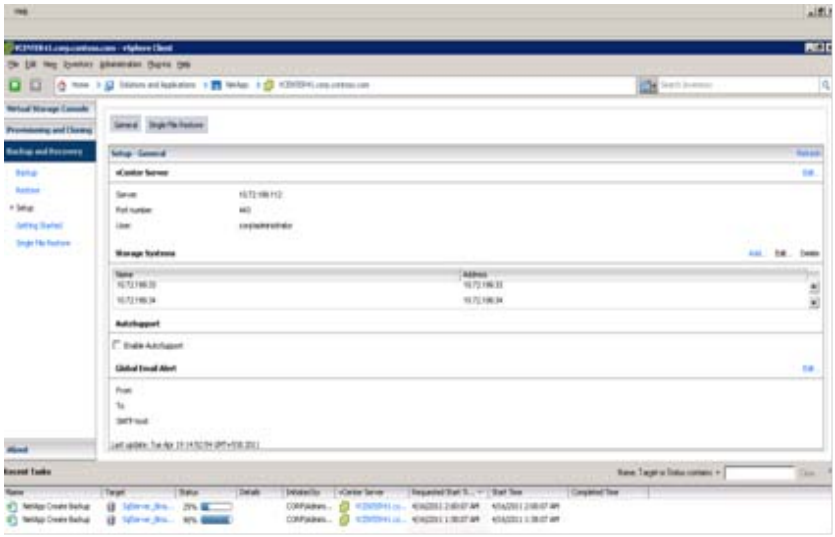
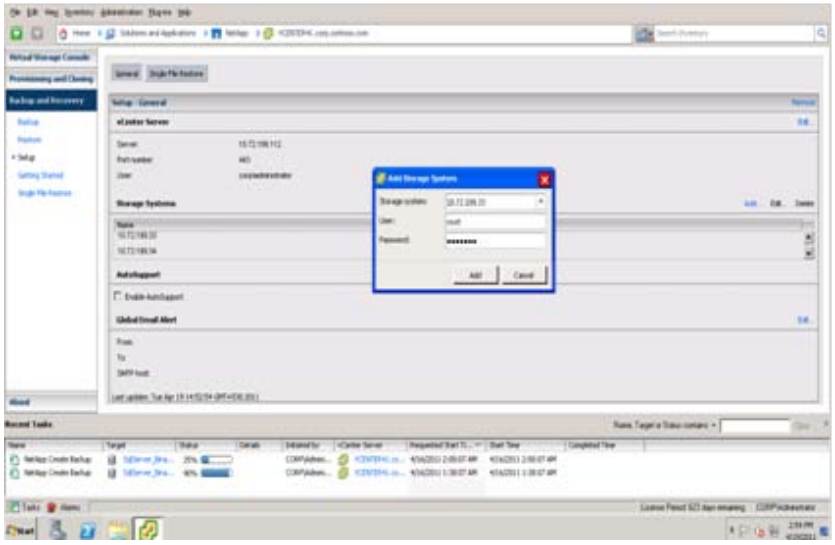
The following section details steps for configuring VMDK LUN for SMSQL using vCenter and VSC.

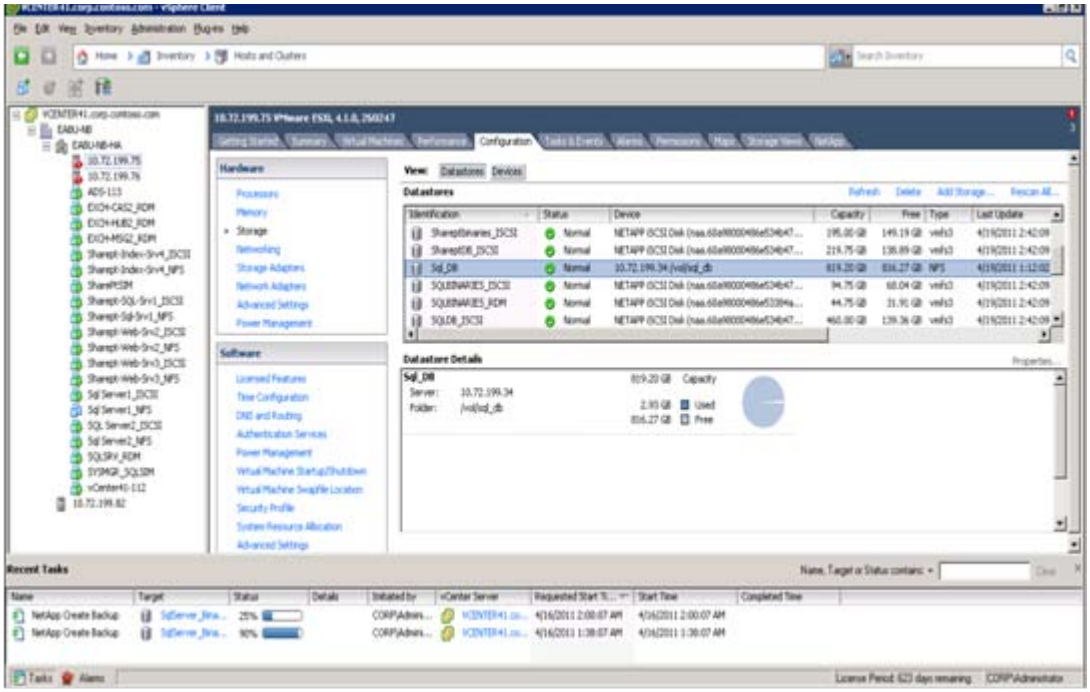
Table 15) Configuring SMSQL on VMDK.

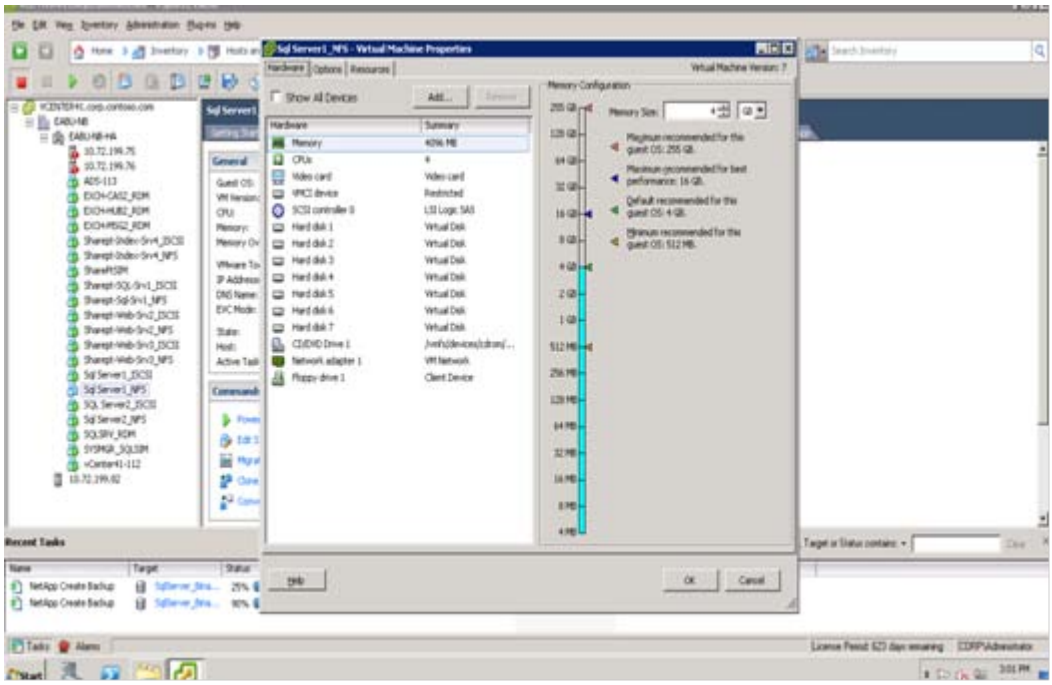
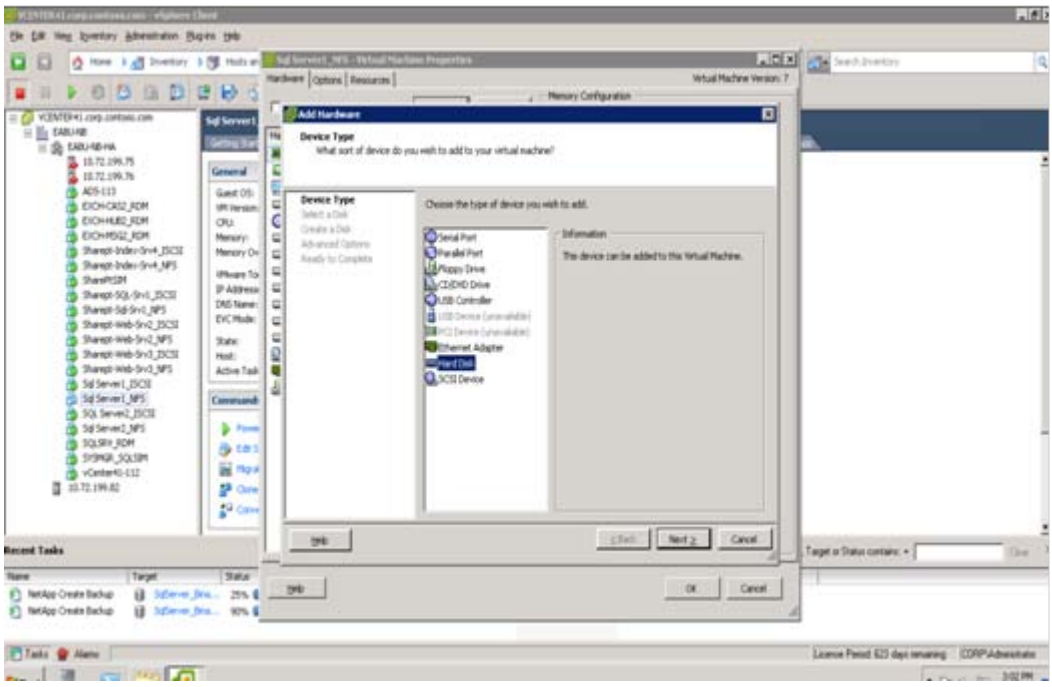
Step	Action
1.	Configuring and managing an ESX server in a production environment requires vCenter 4.1. 

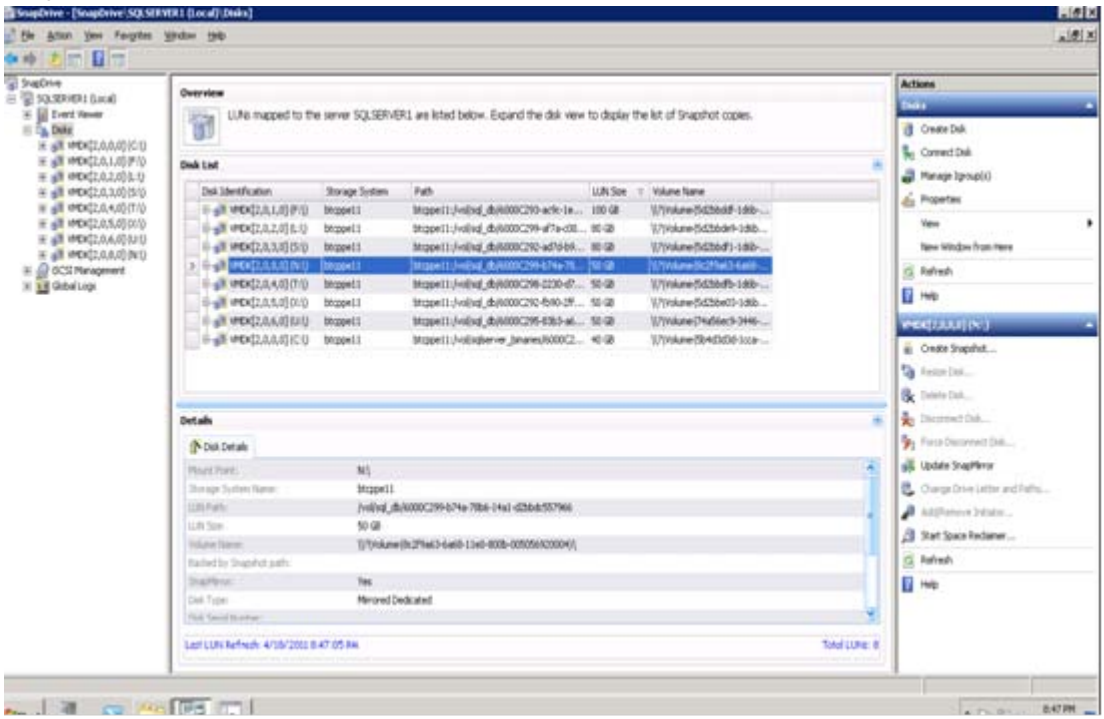
Step	Action
2.	We must add physical servers in vCenter .Right-click on Data Center and click Add Host. Then use the Add Host wizard.  

Step	Action
3.	We must also install the Virtual Storage Console on vCenter. 

Step	Action
4.	After installation we should see the VSC option in vCenter.  We must configure the Virtual Storage Console.  

Step	Action
5.	Create the NFS/iSCSI/FC datastore from vCenter. 

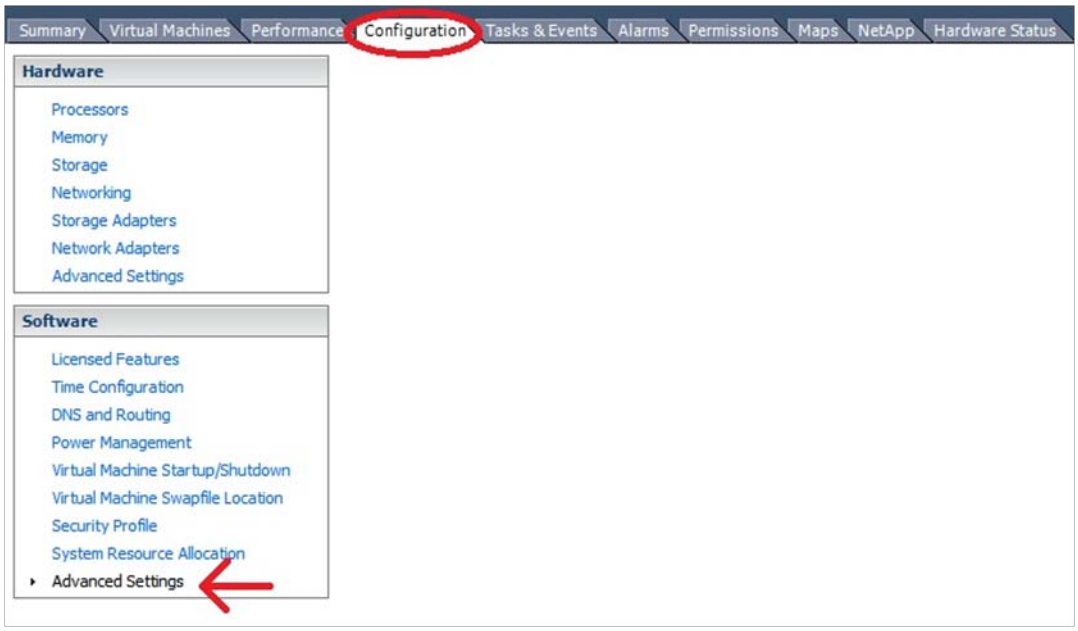
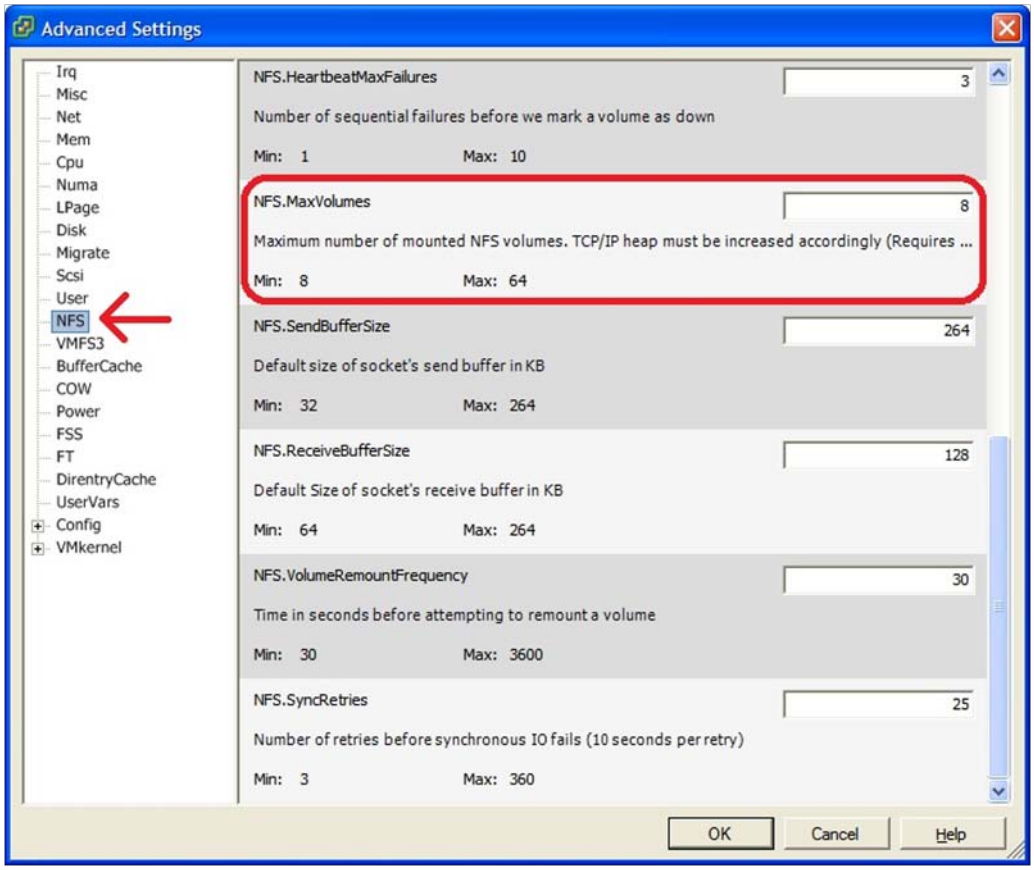
Step	Action
6.	a. Create the VMDK for virtual machines. b. Select the VM. c. Click Edit Setting. d. Click Add. e. Click Hard Disk from Device Type. f. Select Create a New VirtualDisk. (For creating RDM, select Create Raw Device Mapping.) g. Specify the disk size and datastore. Then go to Disk Management and initialize the disk. Format the disk also.  

Step	Action
7.	After the virtual machine is powered on we need to install SnapDrive (6.3 or above) in the virtual machine that was created. During the installation of SnapDrive: a. Select Enable Virtual Center or ESX Server Settings for RDM. b. Select Enable SnapManager for Virtual Infrastructure Configuration Details and specify the IP of the SMVI server (now part of Virtual Storage Console). c. Open SnapDrive for Windows and do a refresh. SnapDrive will detect and show the VMDKs that have been created. 

HOW TO INCREASE THE NFS.MAXVOLUME SETTING

The following procedure describes how to increase the NFS.MaxVolume setting on an ESX server.

Table 16) Changing the NFS.MaxVolumes setting.

Step	Action
1.	To increase or change the NFS.MaxVolumes setting, open the vSphere™ client to the “Hosts and Clusters” view, highlight the ESX host in question, and click on the Configuration tab. Then click on Advanced Settings in the Software box.  

SNAPMANAGER FOR SQL INSTALLATION AND CONFIGURATION

The following sections detail steps for the installation and configuration of SMSQL on the Microsoft SQL Servers. SnapManager for SQL can be installed in either of two modes:

- Interactive mode
- Noninteractive (unattended installation for software packaging) mode

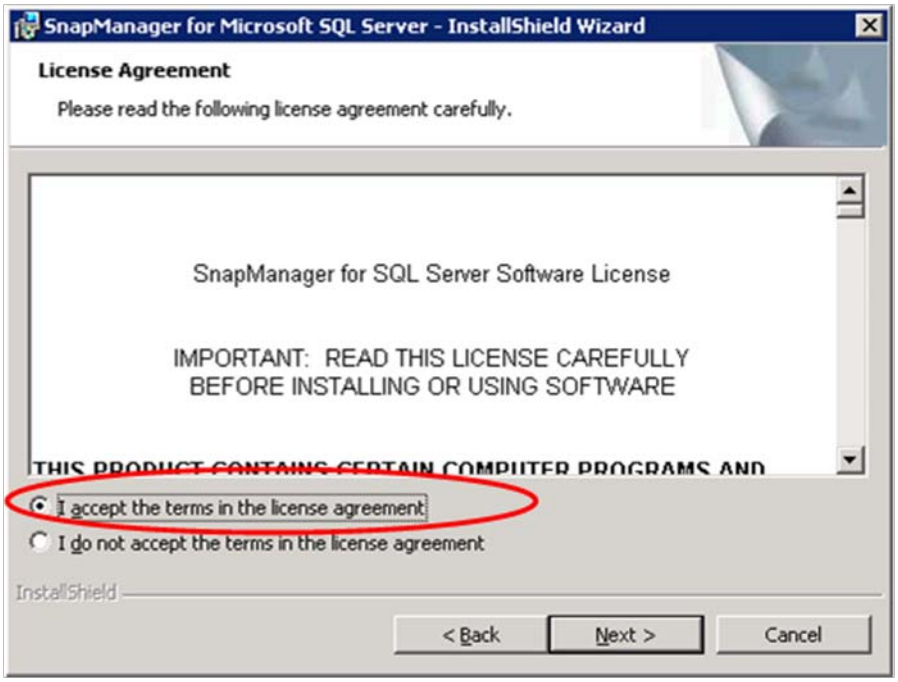
This document does not contain the actual LUN layout for the databases. It covers only the installation of SMSQL and its initial configuration.

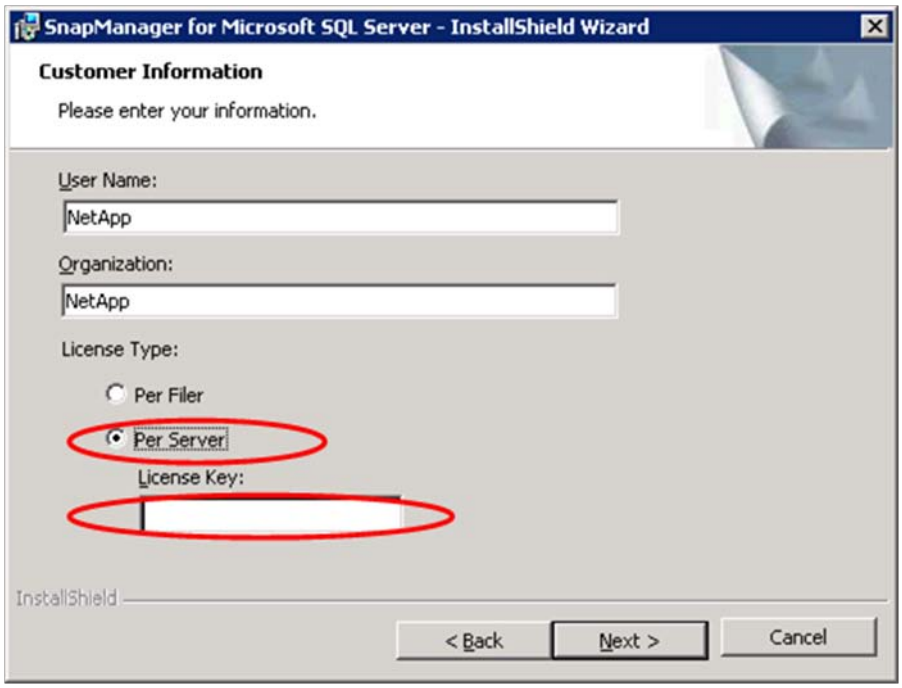
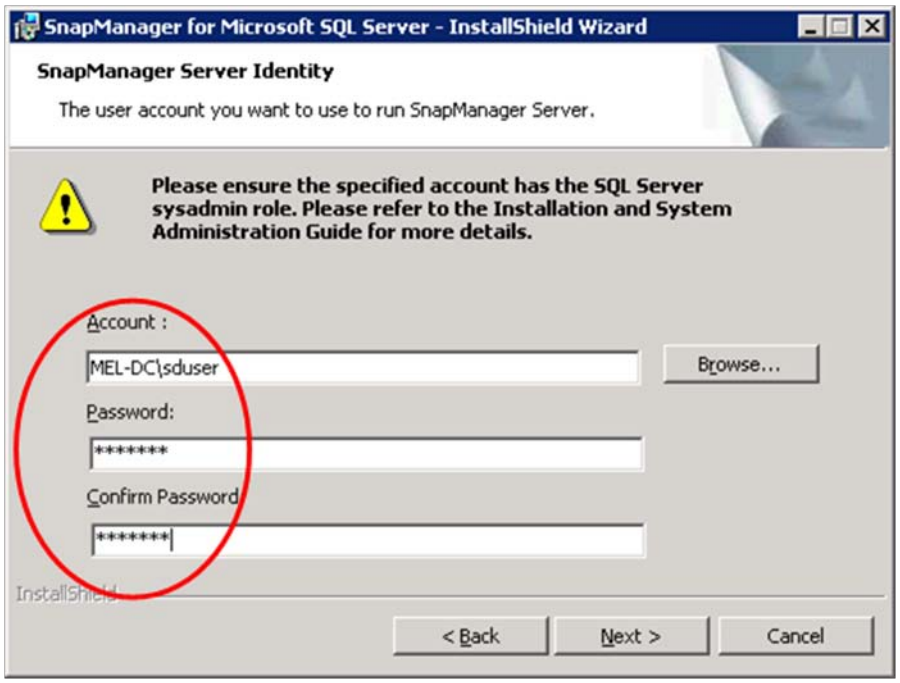
Note: Prior to installing SMSQL, make sure that the Microsoft iSCSI software initiator and SnapDrive software are installed on the Windows servers.

INTERACTIVE MODE FOR INSTALLING SNAPMANAGER FOR SQL

Perform the following steps to install SMSQL on the Windows server dedicated for Microsoft SQL Server:

Table 17) Installing SMSQL on a dedicated Windows server.

Step	Action
1.	Download the SMSQL file from http://now.netapp.com and double-click to open it. Click Next.
2.	Select *I accept the terms in the license agreement*. 

Step	Action
3.	Enter the customer information and select Per Server License as the license type. 
4.	Click Next to install SMSQL on the destination folder.
5.	Enter the SMSQL service account details. 
6.	Click Next. Click Install to begin the installation.
7.	Click Finish to complete the installation.

Step	Action
8.	Repeat the preceding steps on the other MSCS cluster node if it is a Microsoft SQL Server cluster.

UNATTENDED MODE FOR INSTALLING SNAPMANAGER FOR SQL

Perform the steps in Table 18 for the unattended installation of SnapManager for SQL. Using this method, the installation of SMSQL can be packaged and deployed on the Microsoft SQL Servers.

Table 18) Installing SMSQL in unattended mode.

Step	Action
1.	Start the Software Installation utility. If you are installing SnapManager for SQL from a CD/DVD, the executable file is different from the one downloaded from the NetApp Support (formerly NOW®) site. • The SMSQL execution file on the CD is setup.exe. • The downloaded SMSQL execution file is SMSQL5.1.exe.
2.	To install SnapManager for SQL using the command line or through a script for packaging, the following command syntax must be used: **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"** This syntax can be described as follows: **CommandName**: Can be either `setup.exe` or `SMSQL5.1.exe`. **AGREETOLICENSE**: Must be set to YES (this is for license agreement). **USERNAME**: This is optional. **COMPANYNAME**: This is optional. **ISX_SERIALNUM**: This is optional. It is used only when the Microsoft SQL Server-side license is used as opposed to the appliance-side license. **INSTALLDIR**: This is optional. **SVCUSERNAME / SVCUSERPASSWORD / SVCCONFIRMUSERPASSWORD**: These are the SnapManager for SQL service account details; they must be provided. **REBOOT = 0**: This is optional. **/L* TempDirPath\LogFileName**: This is optional. If used it copies all the installation details to the file specified. Example script 1: **E:\setup.exe /v"AGREETOLICENSE=Yes SVCUSERNAME=MKTG2\Administrator SVCUSERPASSWORD=STeeL SVCCONFIRMUSERPASSWORD=STeeL /qb"** Example script 2: **C:\NetApp\downloads\SMSQL5.1.exe /v"AGREETOLICENSE=Yes SVCUSERNAME=MKTG2\Administrator SVCUSERPASSWORD=STeeL SVCCONFIRMUSERPASSWORD=STeeL ISX_SERIALNUM=ABCDEFGHIJKLMN /qb"**

SNAPMANAGER FOR SQL CONFIGURATION

Windows Registry Keys

When used with SnapDrive 4.1 or higher in a clustered environment, SnapManager for SQL 5.1 is capable of restoring databases in the cluster without taking the Microsoft SQL Server clustered server

group offline. It does this by disabling the resource health check monitor for the duration of the LUN restore operation.

To enable this functionality on Windows 2008 servers, configure the following SnapDrive registry values:

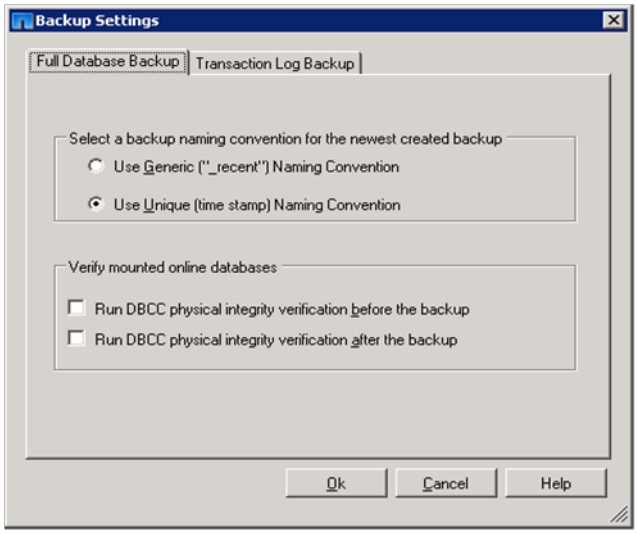
```
HKLM\System\CurrentControlSet\Services\SWSvc\Parameters\LunRestoreWithoutOffline=dword
:0x1
```

On computers running Windows Server 2008, also configure the following SnapDrive registry setting:

```
HKLM\System\CurrentControlSet\Services\SWSvc\Parameters\UseW2K3MaintenanceMode=dword:0
x1
```

SnapManager for SQL Backup Settings

The default backup settings for SnapManager for SQL can be modified to reflect time-based Snapshot copy names as opposed to the recent ones. To change the default behavior, perform the following steps.

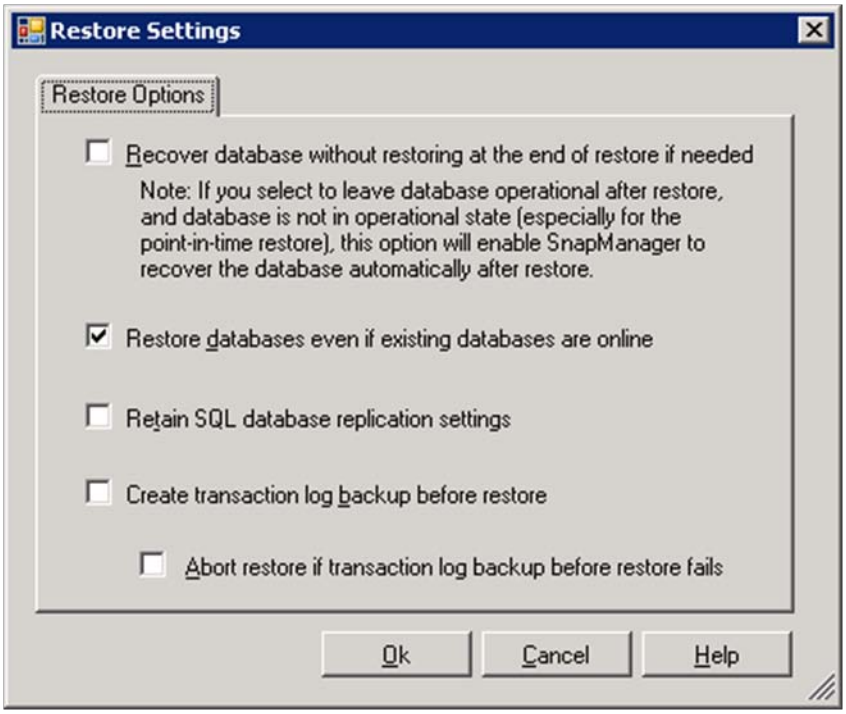
Step	Action
1.	Open SMSQL → Options → Backup Settings.
2.	In the Backup Settings window, make sure that only Use Unique (time stamp) Naming Convention is selected. 

SnapManager for SQL Restore Settings

Make sure the following SnapManager for SQL restore options are selected where SMSQL is installed. NetApp recommends leaving the settings as shown in Figure 17 unless changes are required. To change the restore settings:

1. Open SMSQL → Options → Restore Settings.

Figure 17) SnapManager for SQL restore setting.



2. Changing either of the restore settings can result in the various behaviors listed in Table 19.

Table 19) Restore option descriptions.

Restore Option	Description	Default
Recover database without restoring at the end of restore, if needed.	If this option is selected and the database is not fully operational, SnapManager skips the restore and instead recovers the database to its previous state.	Not selected
Restore databases even if the existing databases are online.	If this option is selected and an existing database is online at the time of the restore operation, SnapManager proceeds with the restore and overwrites the existing database.	Not selected
Retain Microsoft SQL Server database replication settings.	If this option is selected and databases are restored for a Microsoft 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.	Selected

SnapManager for SQL Report Directory Settings

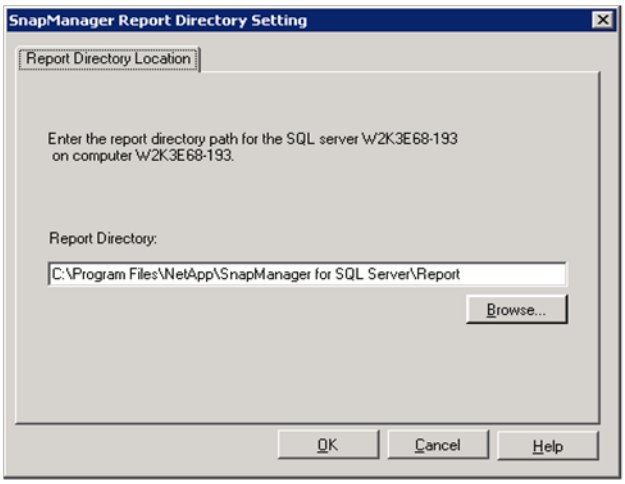
By default the SnapManager for SQL report directory settings are stored where SnapManager for SQL is installed. The location of the SMSQL report directory can be changed for the following reasons:

- Limited space: If there is limited space in the current report directory, it can be moved to a different location with more available disk space.
- Clustered environment: If the Microsoft SQL Server is running on an MSCS cluster, storing the SnapManager reports in the default location (in a directory named Report under the SnapManager

installation directory) does not allow the report directory to be shared between the nodes in the cluster. Consequently the same reports are not seen from different nodes. To avoid these problems, the report directory can be moved to a disk that belongs to the same group as the Microsoft SQL Server virtual server. This change must be performed from every SnapManager for SQL node.

To change the report directory settings, go to SMSQL → Options → Report Settings.

Figure 18) SnapManager report directory setting.



SnapManager for SQL Notification Settings

SnapManager for SQL is configured to send notifications based on reported failures through an email to a recipient or a group. SnapManager can also trigger an AutoSupport message (a feature in Data ONTAP) from the storage controller based on the event so that NetApp Support has all the storage configuration information.

To configure SMSQL notification settings, perform the following steps:

Table 20) Configuring SMSQL notification settings.

Step	Action
1.	Go to SMSQL → Options → Event Notification Settings.
2.	Make the necessary changes. Click OK.