Dell EMC NetWorker Module for Microsoft for SQL VDI

Version 19.2

User Guide

REV 01

November 2019

DELLEMC

CONTENTS

Contents

FIGURES

Figures

TABLES

Tables

Preface

As part of an effort to improve product lines, periodic revisions of software and hardware are released. Therefore, all versions of the software or hardware currently in use might not support some functions that are described in this document. The product release notes provide the most up-to-date information on product features.

If a product does not function correctly or does not function as described in this document, contact a technical support professional.

(i) **Note:** This document was accurate at publication time. To ensure that you are using the latest version of this document, go to the Support website https://www.dell.com/support.

Purpose

This guide contains information about using the NetWorker Module for Microsoft (NMM) 19.2 software to back up and recover SQL Server using the Virtual Device Interface (VDI) technology.

(i) **Note:** The *NetWorker Module for Microsoft Administration Guide* supplements the backup and recovery procedures described in this guide and must be referred to when performing application-specific tasks. Ensure to download a copy of the *NetWorker Module for Microsoft Administration Guide* from the Support website at https://support.emc.com before using this guide.

Audience

This guide is part of the NetWorker Module for Microsoft documentation set and is intended for use by system administrators during the setup and maintenance of the product. Readers should be familiar with the following technologies used in backup and recovery:

- NetWorker software
- Microsoft Virtual Device Interface (VDI) technology

Revision history

The following table presents the revision history of this document.

Table 1 Revision history

Revision	Date	Description
01	November, 2019	First release of this document for the NetWorker Module for Microsoft 19.2 release.

Related documentation

The NMM documentation set includes the following publications:

- *NetWorker Module for Microsoft Release Notes*
- *NetWorker Module for Microsoft Administration Guide*
- *NetWorker Module for Microsoft Installation Guide*
- *NetWorker Module for Microsoft for SQL and SharePoint VSS User Guide*
- *NetWorker Module for Microsoft for SQL VDI User Guide*
- *NetWorker Module for Microsoft for Exchange VSS User Guide*
- *NetWorker Module for Microsoft for Hyper-V User Guide*
- *ItemPoint for Microsoft SQL Server User Guide*
- *ItemPoint for Microsoft Exchange Server User Guide*
- *ItemPoint for Microsoft SharePoint Server User Guide*
- NetWorker documentation set

Special notice conventions that are used in this document

The following conventions are used for special notices:

ⓘ NOTICE Identifies content that warns of potential business or data loss.

ⓘ Note: Contains information that is incidental, but not essential, to the topic.

Typographical conventions

The following type style conventions are used in this document:

Table 2 Style conventions

Bold	Used for interface elements that a user specifically selects or clicks, for example, names of buttons, fields, tab names, and menu paths. Also used for the name of a dialog box, page, pane, screen area with title, table label, and window.
Italic	Used for full titles of publications that are referenced in text.
`Monospace`	Used for: • System code • System output, such as an error message or script • Pathnames, file names, file name extensions, prompts, and syntax • Commands and options

Table 2 Style conventions (continued)

Monospace italic	Used for variables.
Monospace bold	Used for user input.
[ ]	Square brackets enclose optional values.
|	Vertical line indicates alternate selections. The vertical line means or for the alternate selections.
{ }	Braces enclose content that the user must specify, such as x, y, or z.
...	Ellipses indicate non-essential information that is omitted from the example.

You can use the following resources to find more information about this product, obtain support, and provide feedback.

Where to find product documentation

- https://www.dell.com/support
- https://community.emc.com

Where to get support

The Support website https://www.dell.com/support provides access to product licensing, documentation, advisories, downloads, and how-to and troubleshooting information. The information can enable you to resolve a product issue before you contact Support.

To access a product-specific page:

1. Go to https://www.dell.com/support.
2. In the search box, type a product name, and then from the list that appears, select the product.

Knowledgebase

The Knowledgebase contains applicable solutions that you can search for either by solution number (for example, KB000xxxxxx) or by keyword.

To search the Knowledgebase:

1. Go to https://www.dell.com/support.
2. On the **Support** tab, click **Knowledge Base**.
3. In the search box, type either the solution number or keywords. Optionally, you can limit the search to specific products by typing a product name in the search box, and then selecting the product from the list that appears.

Live chat

To participate in a live interactive chat with a support agent:

1. Go to https://www.dell.com/support.
2. On the **Support** tab, click **Contact Support**.
3. On the **Contact Information** page, click the relevant support, and then proceed.

Service requests

To obtain in-depth help from Licensing, submit a service request. To submit a service request:

1. Go to https://www.dell.com/support.

2. On the **Support** tab, click **Service Requests**.

(i) Note: To create a service request, you must have a valid support agreement. For details about either an account or obtaining a valid support agreement, contact a sales representative. To find the details of a service request, in the `Service Request Number` field, type the service request number, and then click the right arrow.

To review an open service request:

1. Go to https://www.dell.com/support.

2. On the **Support** tab, click **Service Requests**.

3. On the **Service Requests** page, under **Manage Your Service Requests**, click **View All Dell Service Requests**.

Online communities

For peer contacts, conversations, and content on product support and solutions, go to the Community Network https://community.emc.com. Interactively engage with customers, partners, and certified professionals online.

How to provide feedback

Feedback helps to improve the accuracy, organization, and overall quality of publications. You can send feedback to DPAD.Doc.Feedback@emc.com.

CHAPTER 1

Overview

This chapter includes the following sections:

Using NMM with Virtual Device Interface API

You can use the NetWorker Module for Microsoft (NMM) software to back up and recover Microsoft SQL Server data. NMM uses Virtual Device Interface (VDI), an API that Microsoft SQL Server provides, to integrate with the SQL Server and enable the NetWorker software to back up and recover SQL Server data.

(i) **Note:** Any references to the Data Domain systems and the Data Domain devices in the product also apply to the PowerProtect Data Domain systems.

When you install NMM, you can run the System Configuration Checker from the Installation wizard. It is recommended that you run the System Configuration Checker to ensure that the setup is correctly configured for backup and recovery. The *NetWorker Module for Microsoft Installation Guide* provides details.

(i) **Note:** If you are a NetWorker Module for SQL Server (NMSQL) user and are migrating to NMM VDI, perform a full backup of the SQL Server data after you install NMM VDI. NMM VDI cannot recover SQL snapshot data backed up with NMSQL.

The following figure describes the backup process that takes place between NMM and the SQL Server using VDI:

1. The user starts the backup process with NMM.
2. The backup command is sent to the SQL Server. This interaction is performed through the VDI API.
3. NMM reads the data from SQL Server and stores it on the NetWorker server.
4. NMM sends the backup status to the NetWorker Server and notifies the user when the backup is complete.

Figure 1 VDI backup process between NMM and SQL Server
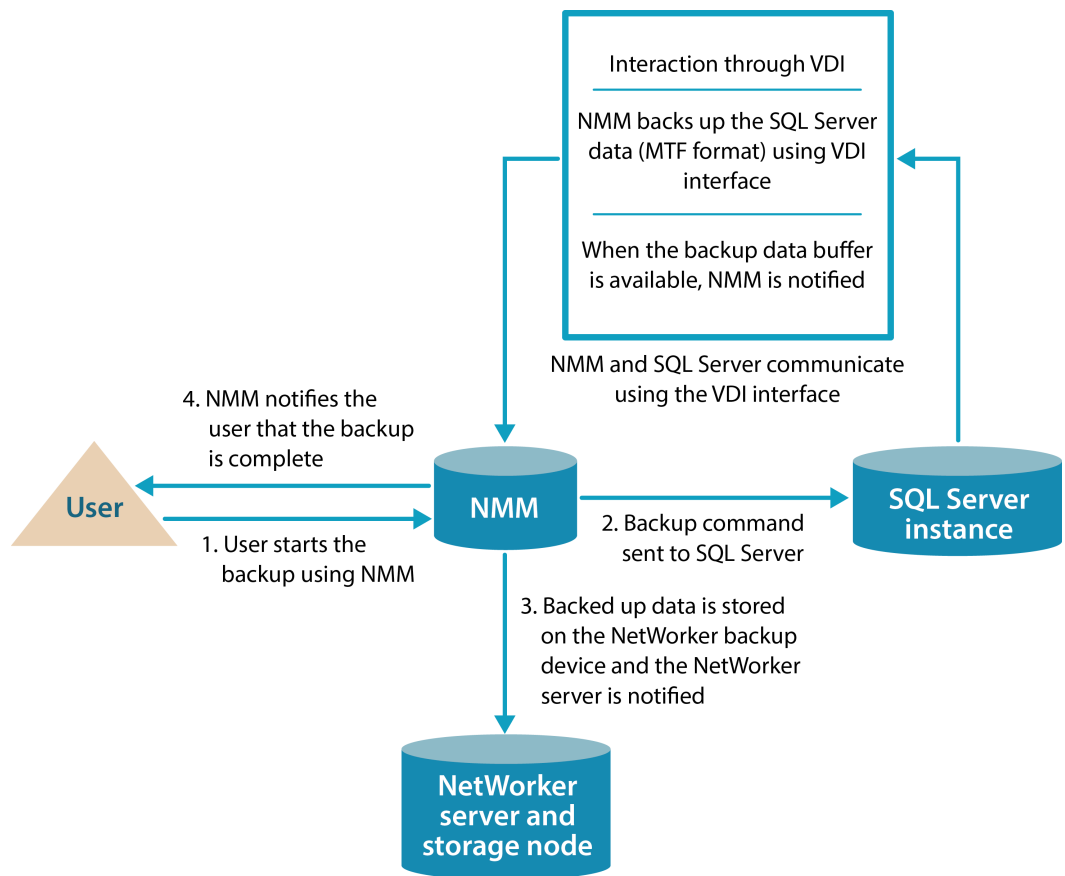
The following figure describes the recovery process that takes place between NMM and the SQL Server using VDI:

1. The user starts the recovery process with NMM.

2. The restore command is sent to the SQL Server. This interaction is performed through the VDI API.

3. NMM reads the data from the NetWorker server and passes the data to the SQL Server using VDI.

4. NMM notifies the user when the recovery is complete.

Figure 2 VDI recovery process between NMM and SQL Server
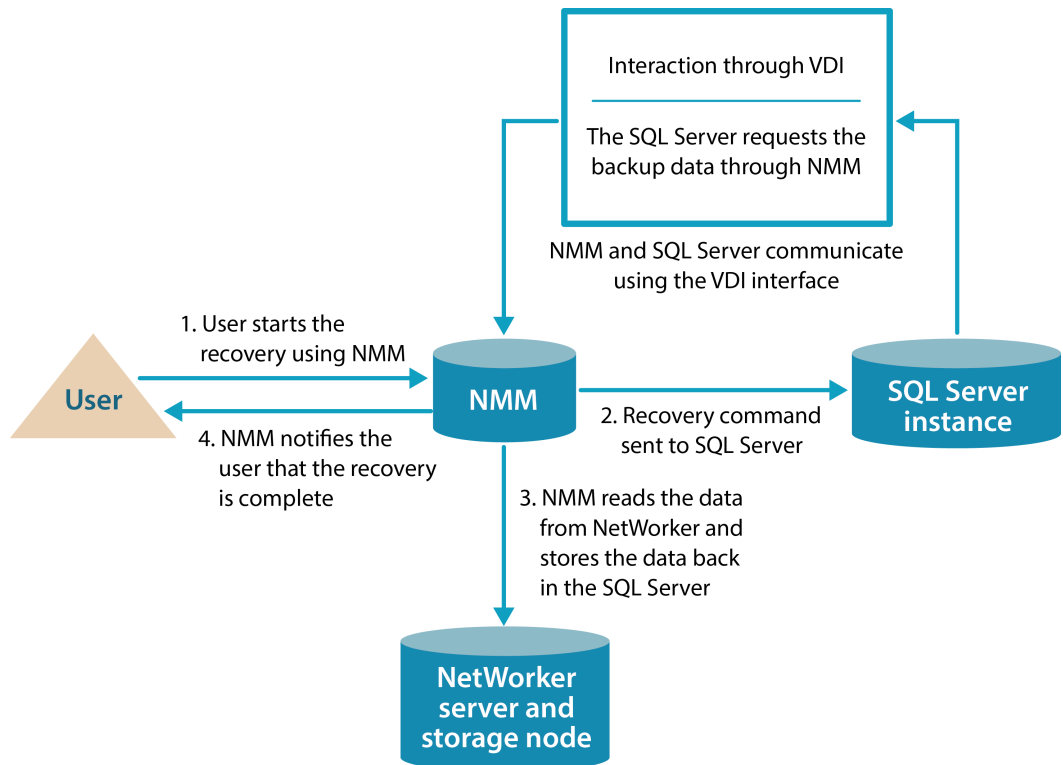
SQL Server backup and restore workflow

This section provides information about SQL Server backup and restore workflows.

Traditional backup workflow (stand-alone backup over a storage node)

During a backup in a traditional environment, processes interact between the NetWorker client (that is, NMM), the NetWorker server, and the SQL Server.

During a traditional backup, the backup starts from the `nsrsqlsv` program, which is started by using one of the following:

- Command prompt
- NMM plug-in for the Microsoft SQL Server Management Studio

The following process occurs in a traditional backup:

1. The `nsrd` program starts the backup `nsrworkflow` on the NetWorker server.
2. The `nsrworkflow` starts the `savegrp` program.
3. The `savegrp` program runs the NMM backup command (`nsrsqlsv`) on the client instead of performing a standard NetWorker save.
4. The `nsrsqlsv` program passes the backup data from SQL Server to the NetWorker server through an X-Open Backup Services application programming interface (XBSA).

The NetWorker server schedules and performs all storage management tasks.

The following figure shows the traditional backup workflow.

Figure 3 Traditional backup workflow
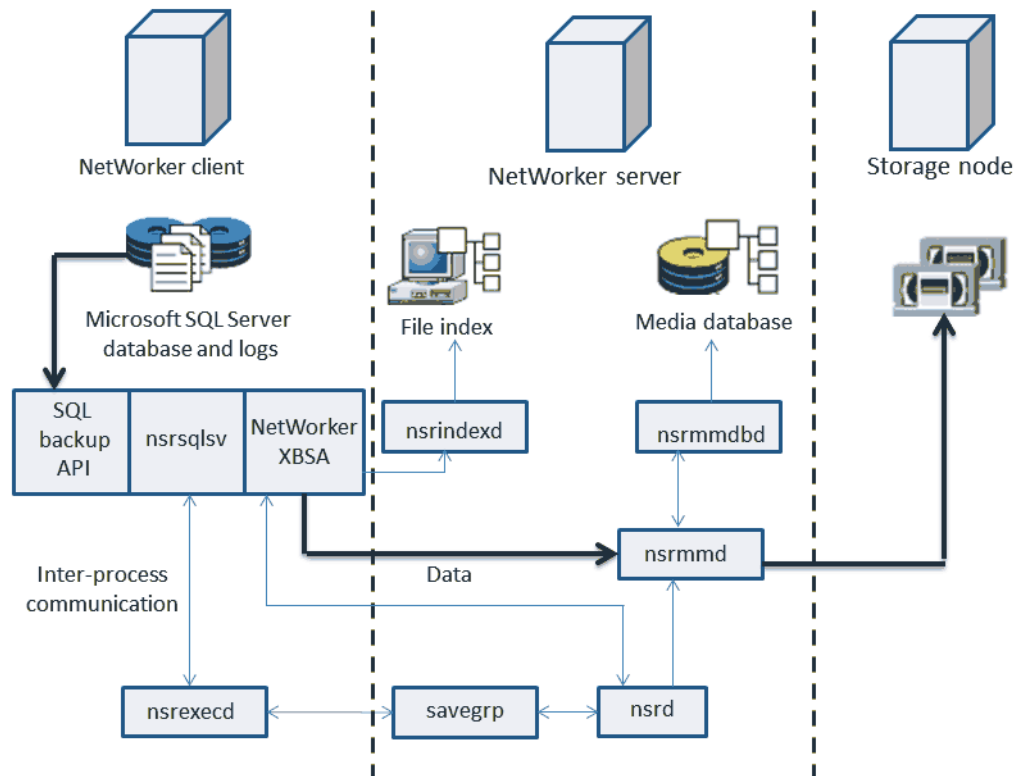
The *NetWorker Administration Guide* provides information about the NetWorker services and operations.

Federated backup workflow (AlwaysOn Availability Group configuration)

During a federated backup, processes interact between the NetWorker client (that is, NMM), the NetWorker server, and the SQL Server.

ⓘ Note: NMM supports federated backups of any SQL Server with support for AlwaysOn Availability Groups (SQL Server 2012 and later).

The backup starts from the `nsrsqlsv` program, which is started by using one of the following:

- Command prompt
- NMM plug-in for the Microsoft SQL Server Management Studio

The following process occurs during a federated backup:

1. The NetWorker server starts the `nsrsqlsv` program in the active node of the Windows cluster (called the coordinator process).

2. The coordinator process queries the SQL Server and detects the Backup Preference and priority from the Availability group, and starts the worker process on the detected preferred node.

3. The backup is configured with and stored under the Windows cluster name on the Availability Group.

The following figure shows the federated backup workflow.

Figure 4 Federated backup workflow
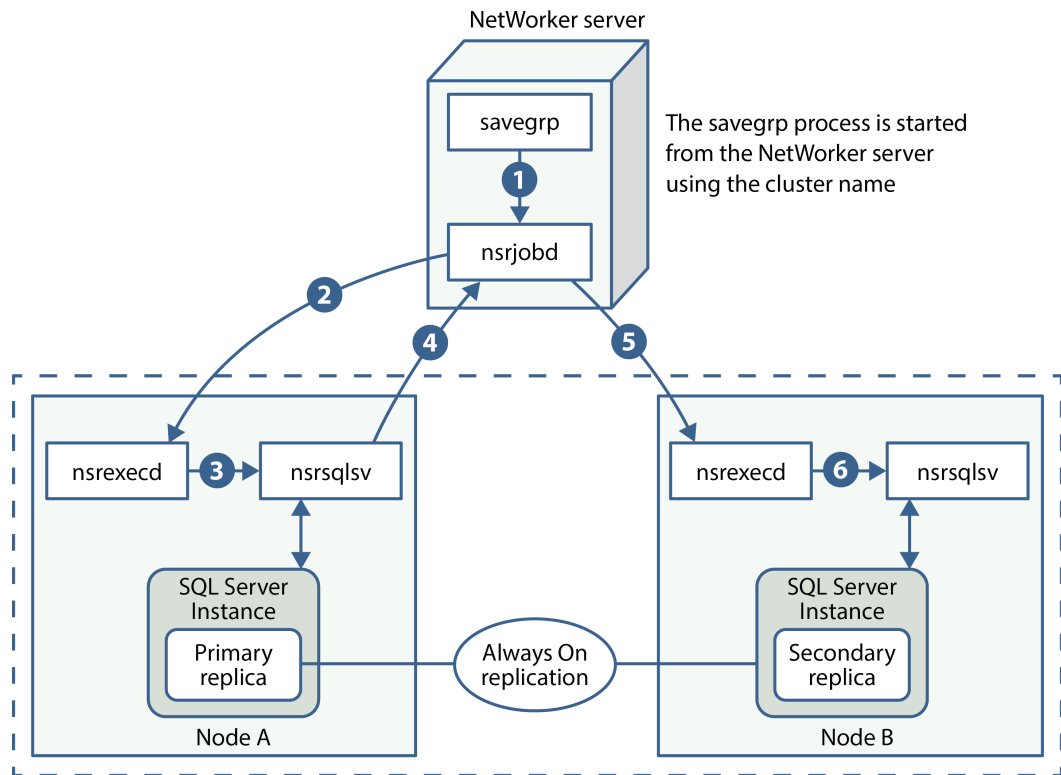
> (i) **Note:** The coordinator process always goes through the `nsrjobd` service to start the worker process on the secondary nodes to ensure that the NetWorker server and `nsrjobd` can communicate with secondary nodes.

Recovery workflow

During a traditional recovery operation that uses a storage node without the Client Direct feature, process interactions occur between the NetWorker client (that is, NMM), the NetWorker server, and the SQL Server.

The following process occurs in a traditional recovery operation:

1. The `nsrsqlrc` program starts the recovery.

2. The NetWorker XBSA API translates the object names that NMM requests into a format that NetWorker understands and forwards the translated object names to the NetWorker server `nsrd` service.

3. The `nsrmmd` media service, contacts the `nsrmmdbd` service to search the NetWorker server's media database for the volumes that contain the requested objects.

4. After the media is mounted, the `nsrmmd` program sends the data through the NetWorker XBSA API to `nsrsqlrc`, which then sends data to the SQL Server.

The following figure shows the traditional recovery workflow.

Figure 5 Traditional recovery workflow
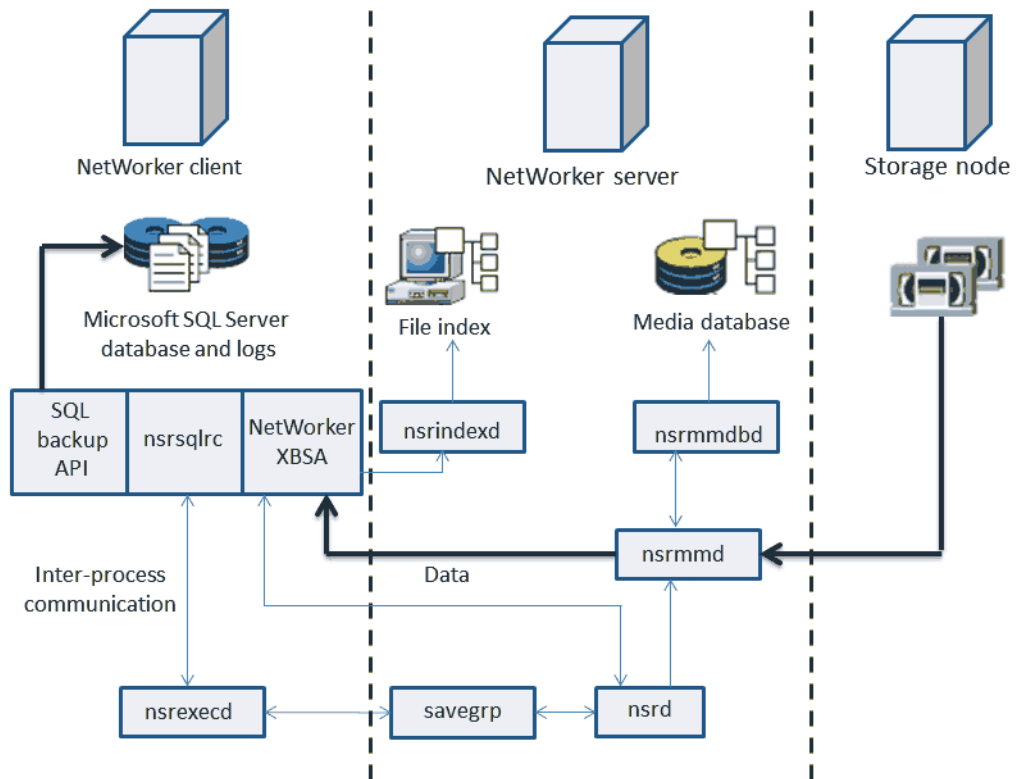
NMM recovery interactions with the SQL Server

When a SQL instance-level recovery occurs, NMM stops and starts the SQL Server and dependent services.

When you want to recover the SQL Server system database types like master and msdb, the `nsrsqlrc` program automatically stops and restarts the SQL Server services appropriately, as follows:

1. Before the recovery process begins, NMM stops the SQL Server and other dependent services.
   When the SQL Server Analysis Services (SSAS) is running, it might use the only available database connection if the SQL Server is in a single-user mode. Stop the Analysis Services before restoring the master database.

2. NMM starts the SQL Server in single-user mode.

3. NMM performs the recovery.

4. After the recovery process finishes, NMM waits for the SQL Server to shut down.

5. For stand-alone and cluster environments, NMM restarts the SSAS.

When you recover a master database, there can be timing issues related to stopping and starting of services. If you are recovering a master database, it is recommended that before you start the recovery, you manually stop all SQL Server services except for the SQL Server.

SQL Server cluster environments

NMM can back up or restore data from an SQL server that is running as a SQL virtual server in a Windows Server Failover Cluster (WSFC). The SQL Server instance that is running in this configuration is called Failover Cluster Instance (FCI).

NMM requires the SQL virtual server name so it can perform the following tasks:

- Connect to the SQL Server instance.
- Accept data from or deliver data to the SQL Server in the cluster, and to initialize the SQL Server VDI.
- Create entries in the NetWorker client file index.

NMM creates index entries under the virtual server name in the NetWorker client file index.

(i) Note: Backup and restore of SQL Server data in a cluster with NMM requires Cluster Client Connection licenses on the NetWorker server host. A separate Cluster Client Connection license is required for each node in the cluster.

NMM detects SQL Server instances

NMM automatically detects all the SQL Servers in a WSFC, including the SQL virtual servers.

NMM detects the SQL Servers only on the active nodes, whenever the following conditions occur:

- You open the NMM plug-in for the Microsoft SQL Server Management Studio.
- You start a backup or restore operation.

Named instances in failover cluster configurations

NMM provides failover cluster support by using the multiple instance features provided in the SQL Server. In a failover configuration, the SQL virtual servers run as either default instances or as named instances. Only one default instance of a SQL virtual server can be installed. Additional SQL virtual servers might be installed as named instances, where each instance name must be unique within the cluster.

Multiple named instances are supported as SQL virtual servers in a cluster configuration. The number of instances that are supported depends on the SQL Server version that is used in the setup. The *NetWorker Module for Microsoft Installation Guide* and the Microsoft SQL Server documentation provide more information.

Each named SQL virtual server instance has the following qualities:

- A unique IP address, network name, and instance name
- Datafile files that are installed on a clustered drive that belongs to the same cluster group as the associated virtual server for that named instance

Active and passive cluster configurations

When the NMM SSMS plug-in GUI is started on the active node, NMM automatically uses the SQL virtual server as the client name. The client name is used for reading or writing to the NetWorker media database and client file index. The NMM SSMS plug-in GUI can be used only on the active node.

When you use the command line interface, use the `nsrsqlsv -A <SQL virtual server>` backup command when the target database is a SQL cluster.

The NMM SSMS plug-in GUI shows all active clusters that are running on the host in the **SQl Server Instance** list.

If each SQL virtual server is running on a different physical node in the cluster, an active or active cluster configuration exists, and no failover occurs, NMM automatically communicates with the SQL virtual server that is running on the same physical node.

Multi-subnet cluster configurations

NMM supports SQL AlwaysOn Availability Groups in a multi-subnet cluster. You can back up and restore the AlwaysOn Availability Group data in a multi-subnet cluster by using the Availability Group Listener client resource. The following sections provide information:

SQL Server backups

This section introduces NMM as a tool to back up an SQL Server.

Types of supported backups

NMM supports manual and scheduled backups of SQL Server data.

Manual (traditional) backup

NMM supports traditional backups, which are often referred to as manual backups. A traditional backup of SQL data can be performed at any time and is independent of any scheduled backup.

NMM supports traditional backup of the following items:

- Database
- File
- File group
- Filestream data
- Transaction log

Also, NMM for SQL Server supports file group differential, file differential, and copy-only backups.

A file group differential backup can reduce both media requirements and recovery time because data is stored across more than one disk or disk partition, so recovery time is reduced. A differential backup can substitute for any logs-only backups performed between the full and differential backups. A full backup must be performed first.

(i) Note: NMM can recover a full backup of SQL Server data (including files and file groups) that were created with NMSQL. However, NMM cannot recover snapshot (PowerSnap based) backups that were created with NMSQL.

Scheduled backup

The most reliable way of protecting SQL data is to ensure that backups of the SQL Server are run at regular intervals, that is, scheduled backups. Scheduled backups ensure that all SQL Server data, including the NetWorker server's client indexes and bootstrap file, is automatically saved. If a disaster occurs, the client indexes and bootstrap file are vital for restoring data to the SQL Server.

Backup limitations

Due to SQL Server behavior, the following limitations apply to SQL Server backups:

- For simple recovery model databases, only full backups (including copy-only full backups) are supported.

- For full backups of a secondary replica, SQL Server supports only copy-only backups.

Backup levels

Learn about the backup levels that NMM supports

(i) **Note:** NetWorker 8.2.x defines backup levels differently than NMM 19.2. If you are using NetWorker 8.2.3 or 8.2.4 server with the NMM 19.2 client, refer to the *NetWorker Module for Microsoft for SQL VDI User Guide* version 8.2 SP1 for information about backup levels.

NMM supports the following three levels of backup.

Full backups

Entire database backup, including all file groups or files in the database.

Cumulative incremental backups

A cumulative incremental backup captures all changes since the last full backup.

Logs-only backups

A logs-only backup that corresponds to a SQL Server transaction log backup. A log file backup cannot be used to recover a database.

A log file backup is used after a database recovery to restore the database to the point of the original failure.

The logs only backup option appears in the NetWorker Management Console. When using the command prompt to perform a logs-only backup, use the `txnlogs -l` command.

(i) **Note:** When a logs-only backup is taken, NMM records two entries in the media database. One entry is the database save set, which lists the backup level as txnlog. The second entry is a cover set entry, which lists the backup level as incr. Both entries are required for cloning and restore operations.

Copy-only backups

You can take a manual SQL Server copy-only backup at any time without affecting the backup schedule or log chain. You can perform copy-only backups at the full or logs-only backup level.

Copy-only backups are not promoted to a different backup level, which allows other backups to run without disrupting the backup chain.

Copy-only full backups are not considered level full backups during promotion. Having a copy-only full backup does not prevent subsequent backups from being promoted to a full backup.

Setting backup levels

NMM enables you to specify backup levels to logs only, cumulative incremental, and full.

The availability of a backup level depends on the type of data selected for backup and any SQL Server settings on those objects, as listed in the following table.

(i) **Note:** NetWorker 8.2.x defines backup levels differently than NMM 19.2. If you are using NetWorker 8.2.3 or 8.2.4 server with the NMM 19.2 client, refer to the *NetWorker Module for Microsoft for SQL VDI User Guide* version 8.2 SP1 for information about backup levels.

Table 3 Backup levels for SQL Server data

SQL Server data objects	Supported SQL Server backup levels		
	Full	Cumulative incremental	Logs only
All databases of SQL default or named instances	Yes	Yes	Yes
Specified databases	Yes	Yes	Yes
All filegroups in specified databases	Yes	Yes	Not applicable
Filestream data in specified databases	Yes	Yes	Yes
Specified filegroups in specified database	Yes	Yes	Not applicable
Specified files in filegroups in specified databases	Yes	Yes	Not applicable

When you perform a logs-only backup for SQL Server data objects, ensure that the SQL Server database options are correctly configured. The Microsoft SQL Server documentation provides more information. Individual items are subject to promotion.

Example strategies for backing up SQL Server data

This section describes example strategies for backing up SQL Server data.

Example 1 Backup strategy one

If the SQL Server manages a significant amount of data, schedule a backup of the databases every 1 to 2 weeks, as shown in the following table.

Table 4 Full backup every 1 to 2 weeks

Fri	Sat	Sun	Mon	Tues	Wed	Thurs
Full	Logs-only	Logs-only	Logs-only	Logs-only	Cumulative incremental	Logs-only
Logs-only	Logs-only	Logs-only	Cumulative incremental	Logs-only	Logs-only	Logs-only
Full	Repeat					

Example 2 Backup strategy two

Another backup strategy is to schedule logs-only backups on several successive days immediately following a full backup, as shown in the following table. This schedule backs up all data that has changed since the previous logs-only backup.

A level 1 cumulative incremental backup can also be scheduled after several days of logs-only backups, as shown in the following table. This schedule backs up all data since the previous full backup.

Example 2 Backup strategy two (continued)

(i) **NOTICE** If a database is read-only, perform a full backup of the database. A read-only database cannot be restored from an existing transaction log backup.

Table 5 Logs-only backup after a full backup

Fri	Sat	Sun	Mon	Tues	Wed	Thurs
Full	Logs-only	Logs-only	Logs-only	Cumulative incremental	Logs-only	Logs-only
Repeat						

Using backup levels

Because it is not practical or efficient to run full backups every day, you can specify different backup levels for scheduled backups. Limiting the frequency of full backups can decrease server load while ensuring complete data protection.

Differences between backup levels

The following table outlines the differences between backup levels.

Table 6 Backup level advantages and disadvantages

Backup level	Advantages	Disadvantages
Full	Fastest restore time.	- Slow backup. - Increases load on client, server, and network. - Uses the most volume space.
Logs only	- Faster backup time than a full backup. - Decreases the load on server and uses the least volume of space. - Enables point-in-time restore.	- Slow restore. - Data can spread across multiple volumes. - Multiple transaction logs can spread across multiple volumes.
Cumulative incremental	- Faster backup time than a full backup. - Captures all changes since the last full backup.	Generally more time-consuming than a logs-only backup (depending on the backup schedule strategy).

Combining data objects to create backup levels

NMM enables the selection of SQL Server data objects in various combinations to create scheduled backups of different levels, as shown in the following table.

Table 7 Creating additional backup levels with data objects

Backup level	Database objects
Full database	To create a level full database backup of the selected databases and their transaction log files, select one or more databases.

Table 7 Creating additional backup levels with data objects (continued)

Backup level	Database objects
Full file or file group	To create a level full file or file group backup of the selected files or file group, but not their transaction logs, select one or more files or one or more filegroups.
Database logs only	To create a database logs only level backup of only the logs only for the selected database, select one or more databases. The SQL database must be previously configured to enable logs only backups.
Database cumulative incremental (level 1)	To create a database level cumulative incremental backup of only the changes that are made to the selected databases since the last full level backup was created, select one or more databases.
File or file group cumulative incremental	To create a file or file group level cumulative incremental backup, select one or more files or one or more filegroups. This backup only includes the changes that are made to the selected files or filegroups since the last full level backup.

Backup promotion

Guidelines for Microsoft SQL Server best practices indicate that a full database backup should be the first step in implementing a restore strategy for a database. In adhering to these guidelines, NMM supports backup level promotion. Backup level promotion is based on data on the NetWorker server and the SQL Server.

(i) Note: NMM does not support backup promotion during copy-only backups.

The following table explains the scenarios which cause backup promotion to occur.

Table 8 Backup level promotion process

Backup item	Requested backup level	Level of promotion	Reason for promotion
Database	Cumulative incremental	Database full	• A full database backup does not exist. • A restore was done after the most recent full database backup. • The last database backup was not performed with NMM. • The database name is "master" or "msdb." • In the case of Always On Availability Groups, the last full backup was performed on a different node. • Either the mirror partner has a more recent backup or its backup status cannot be determined.
	Logs only	Database full	• A full database backup does not exist. • A restore was done after the most recent full database backup.

Table 8 Backup level promotion process (continued)

Backup item	Requested backup level	Level of promotion	Reason for promotion
			• The last database backup was not performed with NMM. • The database name is "master" or "msdb." • In the case of Always On Availability Groups, the last full backup was performed on a different node. • Either the mirror partner has a more recent backup or its backup status cannot be determined. • The database model changed from simple to full or to bulk logged. • The database is using the simple restore model. • The database is in emergency mode.
File/Filegroup	Full	Database full	• A full database backup does not exist. • A restore was done after the most recent full database backup. • The last database backup was not performed with NMM. • The database name is "master" or "msdb." • Either the mirror partner has a more recent backup or its backup status cannot be determined.
	Cumulative incremental	Database full	
	Logs only	Database full	
	Logs only	Filegroup/file full	Logs-only backups of files or filegroups are not supported.

Toggling backup promotion functionality

You can toggle backup promotion for both scheduled and manual backups using the following tools:

*   Scheduled backup:
    *   In the Client Backup Configuration wizard, select **Turn off backup promotion** to turn off backup promotion.
    *   In the Client Properties dialog box, type the `NSR_BACKUP_PROMOTION` application information variable with a valid value.
*   Manual backup:
    *   At a command prompt, use the `BACKUP_PROMOTION` flag with a valid value with the `nsrsqlsv` command.
    *   In the NMM SSMS plug-in, on the **Options** page, select an option from the **Backup Promotion** list.

(i) Note: Consider the following when you disable backup promotion:

*   When backing up the 'msbd' and 'master' databases, the backup level is always set to full and the backup promotion setting is ignored.
*   The first backup of a database is always set to level full and the backup promotion setting is ignored.

- If you disable backup promotion, data loss may occur if the backup chain is broken. For example, if a transaction logs backup is taken with a third party software or SQL native backup tools in between two NMM logs only backups for the same database, NMM may be unable to restore the database using the logs only backup.

The Manual Backups chapter and the Scheduled Backups chapter provide more information about changing backup promotion settings.

SQL Server recovery

This section introduces NMM as a tool to recover an Microsoft SQL Server.

Types of supported recovery

This section lists the types of supported recovery for SQL Server VDI.

ⓘ NOTICE NMM supports recovery of a SQL Server 2012 or later database only after the Always On Availability Group replication has been removed for the corresponding database.

The following table lists the types of recovery for SQL Server VDI in NMM.

Table 9 Types of recovery for SQL Server VDI

Type of recovery	When used	Description
Traditional recovery	For data that was backed up by traditional backup, NMM supports traditional recovery.	Data recovery from a traditional backup can be performed: • At any time with NMM. • By running NMM recover command (nsrsqlrc) from the command prompt. Traditional recovery operations recover files, file groups, databases, and transaction log backups.
Normal recovery	NMM uses the normal restore type as the default.	The normal restore type restores: • The entire set of data that is associated with one or more SQL Server backups, including full, logs only, and cumulative incremental backups. • A file, file group, or a database to the database originally backed up. • Level full, level 1 (cumulative incremental), and level logs only backups in the order required by SQL Server. NMM can back up and restore specified files and file groups. Also, a single file group, or multiple file groups or files, can

Table 9 Types of recovery for SQL Server VDI (continued)

Type of recovery	When used	Description
		be restored from a full database backup.
Copy recovery	A copy recovery is an operation in which data is recovered to a SQL Server host other than the one from which it was backed up. A copy restore from and to the same SQL Server instance can also be done.	The copy recovery type creates a copy of a database by restoring a SQL Server database to a new location, or with a new database name. The copy recovery type makes it easy to duplicate a database that was previously backed up. You can only mark a single item for this operation. Also, you can copy a system database, but you cannot overwrite it. NMM does not support copy recovery of filestream databases.
Flat file recovery	Flat file recovery allows you to restore the backup to a file.	When you perform a flat file recovery, NMM writes the backup to files instead of directly to the SQL database. This feature allows you to restore the recovery files later without access to the network by using the standard T-SQL file restore command.
Granular-level recovery (GLR)	Granular-level recovery allows you to recover SQL server data at the table level.	By using GLR with SQL Server, you can recover individual tables to the production database. This feature reduces the space requirements on local system storage during a restore operation. Depending on the size of the content database, GLR may also reduce recovery time. Granular-level recovery is performed using NMM and ItemPoint for Microsoft SQL Server.

Recovery modes

To recover a database, you must specify a recovery mode. A recovery mode instructs the SQL Server how to interact with the database after the recovery operation completes. For instance, recovery modes can leave the database in an intermediate state, so additional transaction logs can be applied.

The following table shows how the recovery modes correspond to SQL Server database restore options.

Table 10 Recovery modes

Types of recovery mode	Description
Normal	The normal restore mode instructs SQL Server to leave the database in an operational state after the restore completes. This state then enables database reads and writes. The normal restore mode is the default mode NMM uses when restoring a database.
No-recovery	The no-recovery restore mode activates the SQL Server NORECOVERY database restore option for the last stage that was restored. The no-recovery restore mode places the database in a state that cannot be loaded after the restore, but it is still able to process additional transaction log restore operations.
Standby	The standby restore mode activates the SQL Server STANDBY database restore option for the last stage that is restored, which forces the database to be in a read-only state between transaction log restore operations. The standby restore mode provides an undo file for SQL Server to use when rolling back the transactions.
Online	SQL Server provides the ability to perform a restore operation while a SQL Server database is active. The database is completely offline only while the primary file group is being restored. After the primary file group is restored, the database can be brought online while the rest of the file groups are being restored, and then only the data that is being restored is unavailable. The rest of the database remains available during this type of restore. Earlier versions of SQL Server require that you bring a database offline before you restore the database.

Recovery time

Backups can be recovered to a specific time. The recovery time controls which backup data should be reinstated when a database is recovered. The recovery time also controls the portions of a logs only backup that must be recovered when NMM is instructed to discard transactions that are performed after a particular time.

The default or current recovery time for each database comes from the create time of the marked item. By default, the most recent backup is recovered. If the most recent backup is logs only level or 1, dependent backups are recovered first. User-specified recovery times can restore older backup versions or perform point-in-time recovery operations. For example, a point-in-time recovery is specified by using a recovery time that is earlier than the create time of the transaction log backup, but later than the create time of the previous backup.

NMM provides three methods for recovering to a specific time:

- Database backup versions
- Point-in-time recovery of a transaction log (level logs only) backup
- Recovering to a named log mark

Recovery window restrictions

In the **Recovery** window, the rules for marking an item are based on the selected restore type. The normal restore type does not restrict marking in any way. All restorable objects (file, file group, filestream data, database) can be marked. When the copy restore type is chosen, only one database object can be marked. Marking the root SQL Server item is not permitted, the file groups and files of the selected database are automatically marked and restored as part of the full database restore.

The recovery process

A recovery uses the following process.

1. NMM recovers the most recent full backup, and then recovers the most recent differential (level 1) backup (if any).
   If a full database backup is removed from the NetWorker server, and an incremental backup is tried, the recovery fails. NMM checks the SQL Server instance to determine if a full database backup was performed, but does not verify if a full backup still exists on the NetWorker server.

2. NMM recovers all the transaction log backups that ran after the most recent differential backup (or that ran after the last full backup, if there was no differential backup). To correctly recover uncommitted transactions, the SQL Server NORECOVERY mode is specified for all intermediate transaction logs.
   The recovery of the final transaction log specifies the restore mode if STANDBY or NORECOVERY is selected. The default selection is Normal.

   For example, if you selected a restore mode of NORECOVERY, that specification appears in the output for a database restore as follows:

   nsrsqlrc -s *NetWorker_server_name my_database*

   ```
   nsrsqlrc: Restoring database my_database...
   nsrsqlrc: RESTORE database my_database FROM
   virtual_device='BSMSQL' WITH norecovery, stats
   nsrsqlrc: RESTORE database my_database from
   virtual_device='BSMSQL' WITH norecovery (differential)
   nsrsqlrc: RESTORE transaction my_database FROM
   virtual_device='BSMSQL' WITH norecovery
   nsrsqlrc: RESTORE transaction my_database FROM
   virtual_device='BSMSQL' WITH norecovery
   Received 1.0 MB 4 files from NSR server
   ```

Database file relocation restrictions

NMM imposes the following restrictions on database file relocation:

- Only database backups can be relocated. Individual file and file group backups cannot be relocated without relocating the database that contains those files.
- If the configuration of a database has changed since the most recent, level full database back up was created, you cannot relocate the database. Configuration changes include the deletion or addition of files, file groups, or transaction log files.
- A system database might not be the destination database of a relocation.

- The relocation fails if the destination does not have sufficient space to create a database.

SQL Server instance and database names for backup and recovery

SQL Server instance, database, and filegroup names are not case-sensitive. In other words, NMM does not distinguish the difference between upper and lowercase letters in SQL Server instance and database names.

(i) Note: If there are two or more databases with the same name but with different capitalization, such as DB1 and db1, NMM views these databases as the same and by default backs up only one of the databases.

Named and default instances of SQL Server

NMM supports backup and recovery of named and default instances.

NMM supports recovery from the SQL Server default instance or named instances by using a copy restore to any instance of a SQL Server. This support includes recovery operations when the destination server is different from the source.

Each named instance has a unique instance name in the following format:

computerName\instanceName

where:

- *computerName* is the network name of the computer.

- *instanceName* is the logical name of the named instance.

(i) Note: Consider the following information when naming a SQL database or an instance:

- Use unique names.

- Do not use the name that the SQL Server uses to identify itself (`MSSQL:`).

- Do not use the names of SQL instances that you have installed.

Use the following syntax to specify a SQL stand-alone named instance of SQL Server at a command prompt:

`MSSQL$Standalone_Named_Instance:[`*dbName* `...][.`*fgName* `...][.`*fileName* `...]`

An entry of `MSSQL:` for the Save Set attribute during the client resource configuration yields a backup of all databases for the default instance.

When running multiple instances, the `nsrsqlsv` and `nsrsqlrc` commands support specification of only one instance at a time. If save sets for more than one instance are specified, the backup or restore operation fails.

Index entries for stand-alone named instances are created by using the local host on which the instance is running. Index entries for clustered named instances are created with the SQL virtual server name. To differentiate backups for the default instance and named instances, the index name has been extended to logically partition the index for a client.

All running named instances, except clustered instances and the default instance, are maintained in the client file index directory. This named instance directory is created at the end of each traditional backup. Run the `nsrinfo` command after a backup to verify that this directory was created. For example, type:

`%SystemDrive%` `nsrinfo -V -L -n mssql` *client_name*

Supported special characters in database names for NMM backup and recovery

The following table lists the special characters that may be used in SQL Server database names when performing backup and recovery in stand-alone, cluster, Always On Availability Group, and federated configurations.

Table 11 Supported special characters in database names

Special characters		Stand-alone and cluster configurations (Database-level and Instance-level backup and recovery)	Always On Availability Group and federated configurations (availability group level backup and recovery)
~	Tilde	Successful	Successful
-	Hyphen	Successful	Successful
!	Exclamation mark	Successful	Successful
{	Open curly bracket	Successful	Successful
%	Percentage	Successful	Successful
}	Close curly bracket	Successful	Successful
)	Close parenthesis	Successful	Successful
(	Open parenthesis	Successful	Successful
`	Accent grave	Successful	Successful
@	At the rate	Successful	Successful
#	Hash	Successful	Fails ⓘ **Note:** Hash is an availability group identifier.
_	Underscore	Successful	Successful
&	Ampersand	Successful	Successful
^	Caret	Successful	Successful
.	Period	Successful	Successful
\	Backslash	Successful	Successful
'	Apostrophe	Successful ⓘ **Note:** Backup and recovery operations are successful but warnings are displayed when performing recovery.	Successful ⓘ **Note:** Backup and recovery operations are successful but warnings are displayed when performing recovery.
$	Dollar	Fails ⓘ **Note:** Dollar is a SQL instance identifier.	Fails ⓘ **Note:** Dollar is a SQL instance identifier.

Table 11 Supported special characters in database names (continued)

Special characters		Stand-alone and cluster configurations (Database-level and Instance-level backup and recovery)	Always On Availability Group and federated configurations (availability group level backup and recovery)
:	Colon	Fails ⓘ **Note:** Colon is a database name identifier.	Fails ⓘ **Note:** Colon is a database name identifier.

CHAPTER 2

Configuration

This chapter contains the following sections:

Configuring NMM in a SQL VDI environment

This section provides information about configuring NMM in a SQL VDI environment.

Supported Windows Server and SQL Server versions

The *NetWorker E-LAB Navigator* , which is available at https://elabnavigator.emc.com/eln/elnhome, provides the most up-to-date information about supported Windows Server and SQL Server versions.

Migrating from VSS solution to VDI solution for SQL Server data protection

Review this section if you are using an NMM version earlier than NMM 2.4 to recover SQL Server data with VSS technology and want to use NMM 8.2 or later to back up and recover SQL Server data with VDI technology.

- If you are using NetWorker 8.2.3 or 8.2.4 server with NMM 19.2 client, in the **Backup Group Properties** window, clear the **Snapshot** option.
- Assign a backup schedule to the backup.
- In the **Client Properties** dialog box for each client resource:
  - In the **Backup Command** field, type `nsrsqlsv`. For SQL virtual server in a cluster environment, specify `nsrsqlsv -A` SQL virtual server.
  - Leave the **Application Information** field empty.
  - Change the save set to `MSSQL:` for SQL default instance-level backup or `MSSQL:dbname` for database-level backup on a SQL default instance.

Multi-stream Data Domain Boost

NMM supports multi-stream backups for a SQL Server to a Data Domain device. This support leverages the Data Domain Boost feature. SQL Server multi-stream backups over Data Domain Boost enhance the performance by running the backups much faster.

Microsoft SQL Server Always On Availability Group feature

NMM supports federated backups, during which NMM detects the SQL Server preferred backup setting for the Availability Group and performs the backup at the preferred node.

The Microsoft website describes the Always On Availability Group feature in SQL Server 2012 or later and provides detailed information about how to configure the setup for this support. This feature allows multiple replicas of a database.

An availability group is a logical group of databases that has the Always On capability. An availability group is failed over to other nodes as a group, meaning that all the databases that are part of the availability group are failed over together during a failover.

An availability replica hosts each set of availability databases.

Two types of availability replicas exist:

- A single primary replica, which hosts the primary databases.
- One or more secondary replicas, each of which hosts a set of secondary databases and serves as a potential failover target for the availability group. You can configure the following number of secondary replicas depending on the SQL Server version:
  - One to four secondary replicas for SQL Server 2012.

- One to eight secondary replicas for SQL Server 2014 or later.

The secondary replicas can be configured in either synchronous or asynchronous mode.

Consider the following points when you use Always On Availability Groups with NMM:

- To back up secondary replicas with NMM, in the **Availability Group Properties** window, set the **Readable Secondary** configuration option to either **Yes** or **Read-intent only** for each of the primary and secondary replicas. This option allows NMM to connect to the secondary replica to gather information about the secondary replica (for example, database file location, which can be different from the other replicas).

- To restore an Always On Availability Group database, suspend replication before you restore the database. You can use either the NMM Microsoft SQL Server Management Studio plug-in GUI or the T-SQL query window for this task.

It is recommended that you back up an Always On Availability Group with a federated backup workflow, and that you use the Windows cluster name as the client name.

Always On Availability Group failover cluster instance support

NMM supports Always On Availability Groups over two normal SQL clusters. This configuration protects data with both SQL cluster instance-level failover and availability group database-level failover.

ⓘ Note: When you configure a federated backup of an Availability Group that is configured with multiple SQL server instances and one of these instances is a failover cluster instance, use the **Client Properties** dialog box to configure the backup. This situation exists when there is a SQL virtual server instance in the cluster. This configuration is not available from the Client Backup Configuration wizard.

Availability group listeners

NMM supports scheduled and manual federated backups of SQL availability groups through the availability group listener.

Availability group listeners are virtual networks that provide access to an AlwaysOn Availability Group and all of its databases. You can use NMM to backup and recover an availability group using the listener name instead of the cluster name.

Unlike a cluster, when you use the listener name as the backup client name, NMM backs up only the availability group that the listener is configured for.

ⓘ NOTICE When you configure backups of an availability group using the listener name, if you later decide to configure a client for the availability group using either the cluster name or a different listener name, take a full backup with the new client. The same is true in reverse if you decide to change a client using the cluster name to the listener name. NMM cannot restore log-only backups that are taken after the client name changes.

Clusterless availability group listeners

NMM 19.2 and later supports clusterless availability group listeners to perform SQL Server data backups and restores. A clusterless availability group listener enables you to offload secondary copies.

NMM supports SQL clusterless availability group listeners in the following environment and configuration:

- SQL Server 2017 or later
- SQL Server Management Studio 2017 or later
- Windows Server 2012 or later

- NMM 19.2 or later
- Clusters of the type `None`—the value set while creating AlwaysOn Availability Groups
  (i) Note: When you create AlwaysOn Availability Groups, do not set the cluster type to `External`. NMM does not support this type of clusters for backups and restores.

Ensure that you meet the following requirements in a clusterless availability group client environment:

- There must not be any duplicate client IDs.
- Do not use the `nsrclientfix` tool to perform any operations.

Ensure that you perform the following tasks when you configure or use a clusterless availability group listener:

- Assign the IP address of the primary node to the clusterless availability group listener that you use to perform backups or restores.
- In the case of a failover, assign the IP address of the new primary node to the AG listener.

To perform an AlwaysOn Availability Group backup by using a clusterless availability group listener, you must create client resources for the listener and the physical nodes of the AlwaysOn Availability Group. To create the client resources, use the **Client Properties** dialog box of the **NetWorker Administration** window. Configuring client resources for clusterless availability group listener and physical nodes of an AlwaysOn Availability Group on page 90 provides information.

(i) Note: NMM does not support the **Client Backup Configuration** wizard to create a client resource for a clusterless availability group listener.

Also, configure a Data Domain device or an AFTD according to your requirement.

SQL Client Direct to AFTD or DD devices

The NetWorker client software enables clients with network access to Advanced File Type Device (AFTD) or Data Domain devices to send their backup data directly to the devices. This functionality uses the Client Direct feature to allow the client to bypass the NetWorker storage node.

The Client Direct feature is enabled by default, but it can be disabled on each client by clearing the Client Direct attribute on the client resource in NMC. When a Client Direct backup is not performed, a traditional storage node backup is performed instead.

The `nsrsqlsv.log` backup log file displays details about the Client Direct activity for the SQL Server.

Microsoft SQL Server named log marks

Microsoft SQL Server enables enhanced point-in-time restore operations because it allows named log marks to be specified during transactions.

Database applications create named log marks when transactions are performed. The marks enable you to access to specific transaction points in a database transaction log backup. NMM allows you to select a named log mark during a restore operation and restores to the beginning or end of a named log mark during a database restore operation. Restoring data through named log marks is an improvement over point-in-time restore because the time that is associated with restoring to a specific transaction can be more accurately determined.

When a named log mark is created in the SQL Server database, the log mark time is saved to the millisecond. However, NetWorker's time format, which is used to specify point-in-time restore, supports granularity only to the second. If log marks with duplicate names are created within a second of each other, NMM restores to the most recently named log mark.

Managing SQL Server database transaction logs

NMM provides implicit management of the SQL Server database transaction logs.

Implicit management uses SQL database transaction log backups to manage log space. This management can occur when:

- A backup schedule is implemented that is level logs only.

- You run the `nsrsqlsv` command with the `-l txnlog` option.
  - (i) **Note:** If you are using NetWorker server 8.2.x, the backup level is incremental and the `-l incr` option is used with the `nsrsqlsv` command.

Prevent log overflow

In Windows, prevent database logs from overflowing available log space by creating an alert in the SQL Server Performance Monitor. The alert forces a log to backup only when the database's log space reaches a certain capacity (for example, 80% full). An alert is a user-defined response to a SQL Server event. Backups truncate the logs and clear disk space.

Database consistency checks

NMM can provide database consistency checks (DBCC) before a backup operation is performed.

A DBCC examines all tables in the database to detect whether index and data pages are correctly linked and indexes are in proper-sorted order. A DBCC also checks that pointers are consistent and that the data information on each page and page offsets are reasonable. It helps recognize problems early, which prevents problem escalation and possible data loss.

(i) **Note:** DBCC can be performed for database-level, multiple database-level, and instance-level save sets on cluster and stand-alone configurations. For federated configurations, DBCC can be performed for database-level save set, but DBCC cannot be performed for Always On Group level backups.

You can configure a DBCC through the following methods:
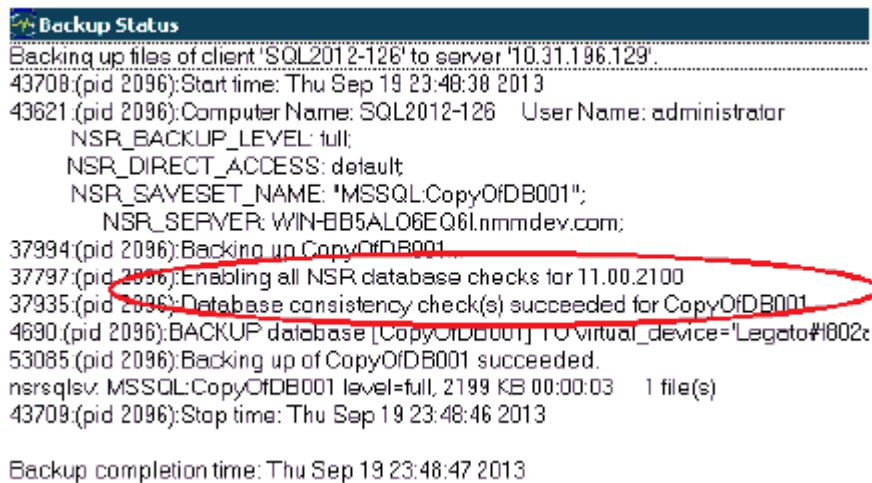
- For manual backups—From the NMM Microsoft SQL Server Management Studio (SSMS) plug-in GUI, in the **NetWorker** window.
  Performing manual backups by using the NMM SSMS plug-in GUI on page 56 provides more information.

- For scheduled and federated backups—In the NetWorker Management Console, in the **Client Properties** dialog box for the SQL Server host client resource:

  - In the **Application Information** field, type
    `NSR_CONSISTENCY_CHECKS=`*`database1,datbase2`* where *database1* and *database2* are the database names for which you want to run a consistency check, for example:
    `NSR_CONSISTENCY_CHECKS=testdb3,CopyOfDB010,test4`
    Ensure that there are no spaces between the database names. For example,
    `NSR_CONSISTENCY_CHECKS=testdb3, CopyOfDB010,test4` is incorrect.

  - To specify that you want to run a consistency check on all the databases in an instance, in the **Application Information** field, type `NSR_CONSISTENCY_CHECKS=ALL`.

- In the **Client Backup Configuration** wizard, select **Perform DBCC consistency check** when you configure the client resource, and then select one of the following:

  - To perform DBCC for all the databases in an instance, select **All**.

  - To perform DBCC for specific databases in an instance, specify the databases for which you want to run the DBCC checks.

- From the command prompt, type the `nsrsqlsv` command with the various backup command options as follows:

- To perform DBCC checks for databases in an instance, type the command option `-j testdb3,CopyOfDB010,test4`.
  Where *testdb3*, *CopyOfDB010*, and *test4* are the names of the databases in the instance. Ensure that there are no spaces between the database names. For example, using `testdb3, CopyOfDB010,test4` is incorrect.

- To perform DBCC checks for all the databases in an instance, type the command option `-j ALL`.

If the DBCC runs successfully for a database, a message appears in the NMM logs or backup output page.

The following figure shows an example of the message that appears when the DBCC is successful.

Figure 6 Message showing DBCC was successful

If DBCC for a database fails, the failure status appears as part of the policy notification and the database backup is omitted. The DBCC failure is reported in the NMM logs or backup output page.

Microsoft hybrid cloud environments

NMM supports SQL Server 2014 and later VDI backups and recoveries in Microsoft hybrid cloud environments.

NMM supports stand-alone, and federated backup workflows for both private and hybrid cloud environments.

Backup and recovery operations in a cloud environment are only supported for onsite databases. The onsite backup workflow is the same as the SQL Server 2012 or later Always On Availability Group workflow.

(i) Note: NMM does not support backup and recovery of SQL Server data directly from the hybrid cloud. If you are using the SQL Server 2016 Stretch Database feature, ensure the data that you must back up and recover is onsite, otherwise the operation fails.

Transparent data encryption

Microsoft SQL transparent data encryption (TDE) is a feature that performs realtime I/O encryption and decryption of the data and log files.

TDE uses a database encryption key (DEK), which is stored in the database boot record for availability during recovery. Encryption of the database file is performed at the page level. The pages in an encrypted database are encrypted before they are written to disk and decrypted when

read into memory. When using this feature, ensure that the certificate and private key are backed up with the encrypted data.

(i) **Note:** NMM does not support third party transparent data encryption for SQL VDI.

Microsoft SQL Server 2008 and later introduce the TDE database-level encryption feature. This feature provides protection for the entire database at rest, without affecting existing applications. NMM supports SQL data encryption at the cell level, at the full database level by using TDE, or at the file-level with encryption options provided by Microsoft.

The Microsoft SQL Server product documentation provides more information about TDE, enabling data encryption, and protecting the encryption keys.

(i) **Note:** When enabling TDE, back up the certificate and the private key associated with the certificate. If the certificate becomes unavailable or if the database is restored on another server, backups of both the certificate and the private key must be available to open the database.

Setting the MAXTRANSFERSIZE environment variable

MAXTRANSFERSIZE is a data transfer option that specifies the largest unit of transfer in bytes to be used between SQL Server and the backup media or VDI backup application. This option applies to both backup and recovery operations.

About this task

The MAXTRANSFERSIZE value is set through the NSR_VDI_MAXTRANSFERSIZE environment variable. Setting the NSR_VDI_MAXTRANSFERSIZE variable determines the MAXTRANSFERSIZE value as follows:

MAXTRANSFERSIZE= 1024 x *NSR_VDI_MAXTRANSFERSIZE*

The valid values of the NSR_VDI_MAXTRANSFERSIZE variable range from 64 to 4096. The resulting values of the MAXTRANSFERSIZE variable range from 65,536 bytes to 4,194,304 bytes.

Use the following procedure to change the MAXTRANSFERSIZE from the default value of 4,194,304 bytes.

Procedure

1. Open **Control Panel** > **System**.

2. Click **Advanced system settings**.

   The **System Properties** window opens.

3. Click **Environment Variables…**

   The **Environment Variables** window opens.

4. In the **User variables** section, perform one of the following:

   - If an NSR_VDI_MAXTRANSFERSIZE variable is not listed, click **New** to create and define the NSR_VDI_MAXTRANSFERSIZE variable.

   - If an NSR_VDI_MAXTRANSFERSIZE variable is listed, click **Edit** to edit the variable.

5. Click **OK**.

Results

After this parameter is set in the registry, subsequent NMM backups use the registry setting. For SQL cluster environments, set the key on all cluster nodes where backups might run.

Configuring permissions to perform NMM backup and recovery of Microsoft SQL Server

Review the privileges that are required to perform Microsoft SQL Server backup and recovery operations with NetWorker Module for Microsoft (NMM).

Access privileges for backup and recovery

NMM for Microsoft SQL Server VDI requires that the user starting backup and recovery operations is assigned certain privileges from the Microsoft SQL Server, the Windows application host, and the NetWorker server.

These privileges are required for both scheduled and manual backups:

- For manual backups, the user starting the backup must be granted the required privileges.
- For scheduled backups, the remote user who is assigned to the client resource must be granted the required privileges.

Microsoft documentation provides additional information on how to configure user accounts.

Table 12 Access privileges required for backup and recovery operations

Microsoft SQL Server user roles	Windows user groups	NetWorker user roles
Assign the user to the following server roles with the Microsoft SQL Server Studio Management (SSMS) GUI: • sysadmin • public Assign the NT AUTHORITY \SYSTEM user to the following server roles with the Microsoft SSMS GUI: • sysadmin • public	Assign the user to the following user groups on the Windows application host on all SQL nodes: • Backup Operators • Administrators ⓘ Note: The user must be a domain user for failover cluster or availability group backups. You can use a local user for standalone backups.	Assign the user to the Operators role in the NetWorker Administration GUI. The *NetWorker Administration Guide* provides more information about assigning NetWorker User roles.

Assign SQL server roles for backup and recovery operations

Assign the required privileges on the SQL Server to perform SQL Server backup and recovery operations.

Before you begin

A user account must exist on the SQL Server.

Procedure

1. Launch the SQL Server Management Studio.
2. In the **Connect to Server** window, specify the details for the SQL Server that you want to backup and recover. Ensure that you select **Windows Authentication**.
3. From the **Object Explorer** pane, expand the folder for the SQL Server that you want to backup and recover, then expand the **Security** folder, and then expand the **Logins** folder.

4. Right-click the name of the user that you want to assign backup and recovery privileges to, and then click **Properties**.

   The **Login Properties** window appears.

5. On the **Server Roles** page, ensure that **sysadmin** and **public** are selected, and then click **OK**.

   Figure 7 Assigning SQL Server privileges
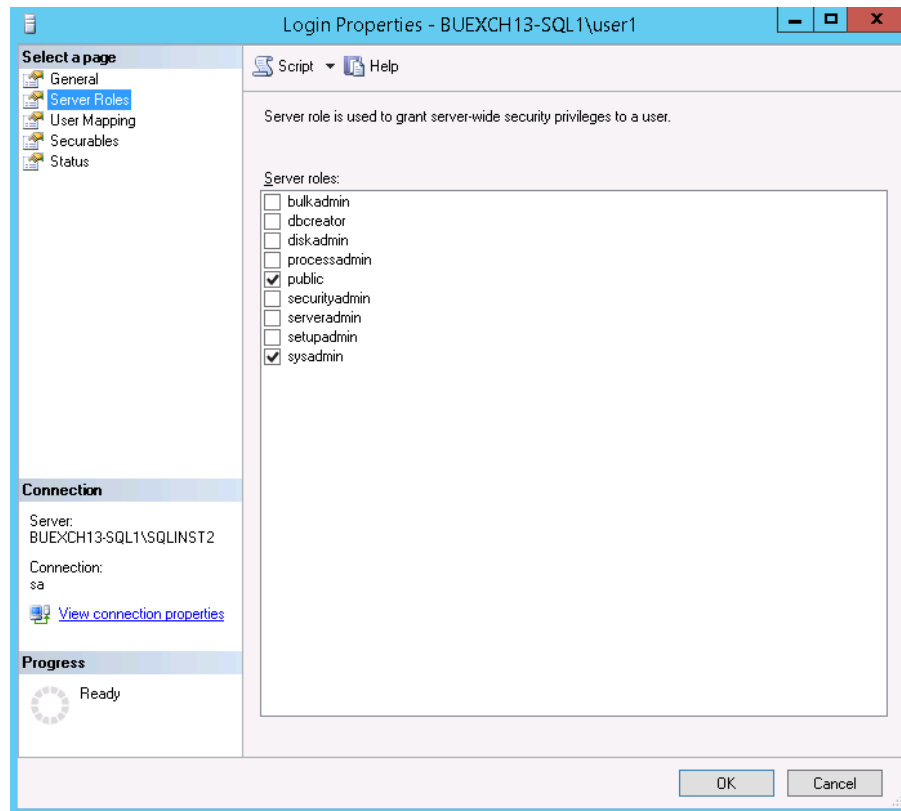
6. Return to the **Logins** folder.

7. Right-click the "NT AUTHORITY\SYSTEM" user, and then click **Properties**.

   The **Login Properties** window appears.

8. On the **Server Roles** page, ensure that **sysadmin** and **public** are selected, and then click **OK**.

Assigning Windows user privileges for backup and recovery operations

Assign the required user privileges on the Windows application host to perform SQL Server backup and recovery operations with NMM.

There are two types of users of which access privileges can be modified:

- Local User
  This type of user is used for standalone server and Always On Availability Group databases.

- Domain User
  This type of user is used for Always On Availability Group and Failover Cluster Instance databases.

Assign user privileges to a Local User

Local user privileges are modified using the **Local Users and Groups** window.

Procedure

1. On each SQL Server node that you want to back up and recover, click **Start** > **Programs** > **Administrative Tools** > **Computer Management** > **Local Users and Groups**.

2. In the left pane, under **Local Users and Groups (Local)**, perform one of the following action sequences.

   - To assign privileges to an existing user, use the following steps:

     a. Click **Users**.

     b. From the list of Local Users, right-click the user, and then click **Properties**.

   - To assign privileges to a new user, use the following steps:

     a. Select and right-click **Users**, and then click **New User**.

     b. In the **New User** window, specify the details for the new user, and then click **Create**.
        The user appears in the list of Local Users in the **Users** folder.

     c. Right-click the newly created user, and then click **Properties**.

   The **User Properties** window appears.

3. On the **Member Of** tab, add the user to the **Backup Operators** and **Administrators** user groups, and then click **Apply**.

4. Click **OK**.

Assign user privileges to a Domain User

Domain user privileges are modified using the **Active Directory Users and Computers** window.

Procedure

1. On the domain controller, click **Start** > **Programs** > **Administrative Tools** > **Active Directory Users and Computers**.

2. In the left pane, under **Active Directory Users and Computers**, expand the Active Directory domain name, and then perform one of the following action sequences:

   - To verify that an existing user is a domain user, use the following steps:

     a. Click **Users**.

     b. From the list of Active Directory users, right-click the user, and then click **Properties**.

     c. On the **Member Of** tab, verify that the user is listed as a member of the Active Directory domain.

   - To create a new domain user, use the following steps:

     a. Select and right-click **Users**, and then click **New** > **User**.
        The **New Object - User** window appears.

     b. In the **New Object - User** window, select the Active Directory domain for the account and specify the details for the new user, and then click **Next**.
        The user appears in the list of Active Directory users in the **Users** folder.

3. In the **Users** folder, right-click the user that you want to use for NMM backup and recovery, and then click **Properties**.

4. In the user **Properties** window, on the **Member Of** tab, click **Add**.

5. Add the user to the **Backup Operators** and **Administrators** user groups, and then click **Apply**.

   The user groups are listed on the **Member Of** tab.

   Figure 8 Adding a user to Windows User Groups
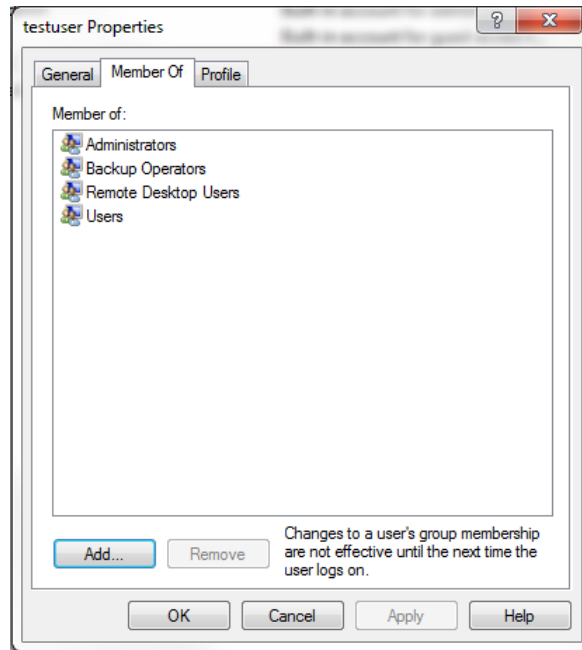
Enable user access for NMM when User Access Control (UAC) is used

Grant the Windows "Log on as a batch job" privilege to the remote user that performs NMM operations. This privilege allows the user to log in with a privileged security token

About this task

ⓘ **Note:** These steps must be performed on each node in the SQL Server cluster.

Procedure

1. Open the Local Security Policy (secpol.msc) on the client

2. Browse to **Local Policies** > **User Rights Assignment**.

3. Add the Backup Operators and Administrators user groups to the **Log on as a batch job** privilege.

4. Click **OK**.

Configuration

CHAPTER 3

Graphical User Interfaces

This chapter includes the following sections:

User interfaces for backup and restore

This section describes the graphical user interfaces (GUIs) where you can perform backup and restore operations.

- The NetWorker Administration GUI on the NetWorker server: Start NMC on the NetWorker server and open the NetWorker Administration GUI to configure, perform, monitor, and report on scheduled backups.

- NetWorker window in the NMM Microsoft SQL Server Management Studio plug-in GUI: Start the NMM Microsoft SQL Server Management Studio (NMM SSMS) plug-in GUI to view the **NetWorker** window. You can use the **NetWorker** window to perform manual backup and restore operations.

The following table summarizes the locations from which you can start backup operations, and the backup levels that are supported for each interface.

Table 13 Where to start backup operations

Backup type	Backup started from	Available backup levels		
		Full	txnlog	Diff
Scheduled	The NetWorker Administration GUI on the NetWorker server	Yes	Yes	Yes
Manual	Command prompt on the SQL Server, which is the client	Yes	Yes	Yes
	In the NMM SSMS plug-in GUI, in the **NetWorker** window	Yes	Yes	Yes

The following table summarizes the locations, from which you can start restore operations.

Table 14 Where to start restore operations

Restore type	Restore started from
Full	In the NMM SSMS plug-in GUI, in the **NetWorker** window

NMM Microsoft SQL Server Management Studio plug-in GUI

You can use the NMM Microsoft SQL Server Management Studio (SSMS) plug-in GUI to perform manual backup and restore operations.

Perform backup and restore operations by using the **NetWorker** window, with the **Backup**, **Database Restore**, and optional **Table Restore** tabs in the **NetWorker** window.

To use this feature, select the NMM SSMS plug-in GUI option during NMM installation. To enable the **Table Restore** tab, select the SQL Granular Level Recovery option whenever you install NMM.

(i) **Note:** If you have not installed the SSMS option and want to launch the NMM SSMS plug-in GUI, navigate to the `<install_path>\nsr\bin\` folder, and then double-click the nwssmsaddinapp.exe file.

Views of the NMM SQL Server Management Studio plug-in GUI

The **NetWorker** window in the NMM SMSS plug-in GUI has two main tabs and a third optional tab with multiple pages in each.

The following figure shows the **NetWorker** window in the NMM SMSS plug-in GUI.

Figure 9 The NetWorker window in the NMM SMSS plug-in GUI
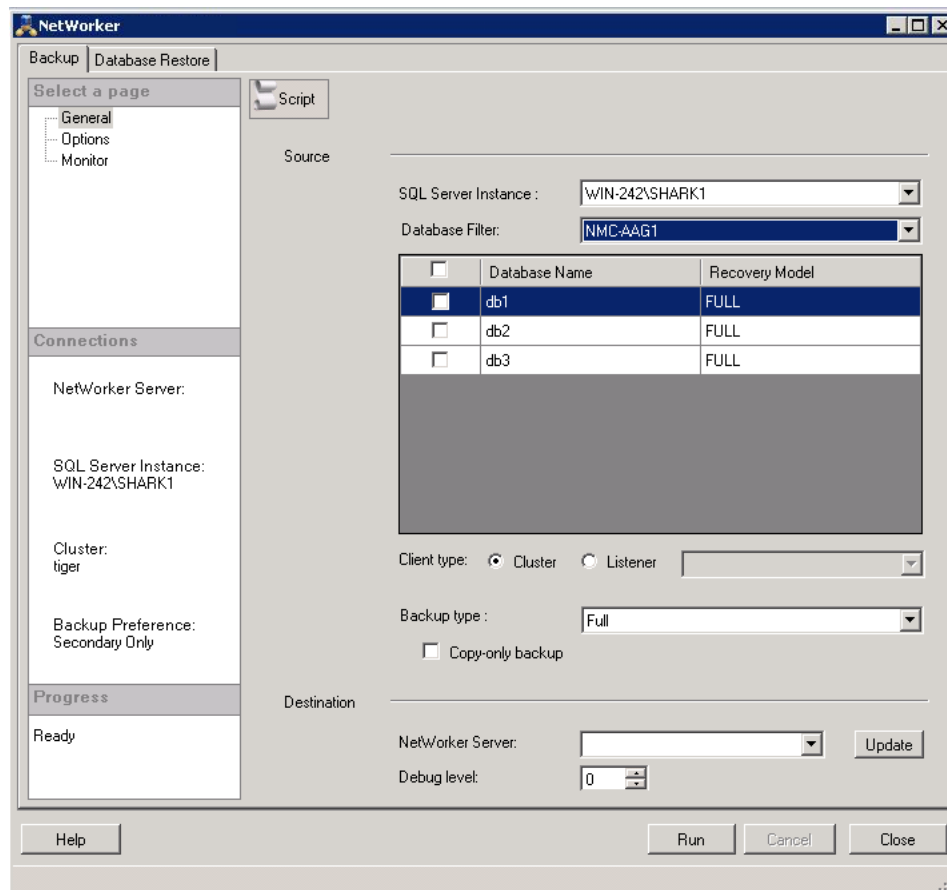
Backup tab

In the **NetWorker** window, click the **Backup** tab to see the following pages which must be used to perform backup operations:

- General
- Options
- Monitor

Database Restore tab

In the **NetWorker** window, click the **Database Restore** tab to see the following pages which must be used to perform recovery operations:

- General
- Files/Filegroups
- Options
- Monitor

Table Restore tab

If you installed the SQL Granular Recovery functionality, from the **NetWorker Backup** window select the **Table Restore** tab to see the following pages which must be used to perform table-level recovery:

- General
- Options
- Monitor

CHAPTER 4

Manual Backups

This chapter includes the following sections:

Manual backup overview

You can start a manual backup of SQL data at any time. A manual (unscheduled) backup can be immediately started.

Manual backups are generally performed under special circumstances, such as NMM setup.

Before performing a manual backup, ensure that the user starting the backup is assigned the required user privileges as defined in the section Access privileges for backup and recovery on page 44.

For manual backups of SQL Server cluster environments, the backup operation must be run from the active host in the cluster.

The following combinations of data objects can be backed up manually with NMM:

- The entire SQL Server storage hierarchy
- One or more entire databases
- One or more file groups
- One or more files in a file group
- A heterogeneous collection of files, file groups, and databases
- Transaction log backups

(i) Note: Filestream data, which are stored in SQL Server 2008 or later databases, is displayed in the backup window as a single file group folder without subordinate objects.

The storage hierarchy is defined as the database storage components the SQL Server Storage Engine exposes to third-party backup vendors. The storage components include files, file groups, databases, and transaction logs.

When performing a manual full backup of a file or file group, also perform a database logs-only level backup to maintain the validity of the transaction log. You can perform logs-only backups only through the NMM Microsoft SQL Server Management Studio plug-in GUI, in the **NetWorker** window.

(i) Note: Manual backups of SQL data do not back up the client indexes and bootstrap file. If you are using NetWorker 8.2.3 or 8.2.4 server with NMM 19.2, follow the instructions provided in the "Backing up client indexes and a bootstrap file" section of the *NetWorker Module for Microsoft for SQL VDI Release User Guide* version 8.2 SP1 to back up the client indexes and bootstrap file. NetWorker server 9.x and later have a built-in protection policy to back up the client indexes and bootstrap file.

Federated backup preferences for Availability Group databases

You can configure the SQL Server backup preference for Availability Group databases by using either the Microsoft SSMS GUI or the Transact-SQL command.

NMM supports federated backups, during which NMM detects the SQL Server preferred backup setting for the Availability Group and performs the backup at the preferred node. The database administrator can set the backup priority for the Availability Group or a database in the Availability Group and nominate a particular replica for the backup. This feature improves backup performance.

SQL Server 2012 or later provides the following options that determine where backups run for a specific Availability Group replica:

- **AUTOMATED_BACKUP_PREFERENCE**—Specify any one of the following options:

  - **PRIMARY**—Specifies that the backups will always occur on the primary replica. This option is useful if you need backup features, such as creating differential backups, that are not supported when backup operations run on a secondary replica.

  - **SECONDARY_ONLY**—Specifies that backups will never be performed on the primary replica. If the primary replica is the only replica that is online, the backup will not occur.

  - **SECONDARY** (Prefer Secondary)—Specifies that backups should occur on the secondary replica. If the primary replica is the only replica online, the backup is performed using the primary replica. This is the default option.

  - **NONE** (Any replica)—Specifies that you prefer that backup jobs ignore the role of the availability replicas when choosing the replica on which to perform backups. Note backup jobs might evaluate other factors such as backup priority of each availability replica in combination with its operational state and connected state.

- **BACKUP_PRIORITY =n**—Specifies your priority for performing backups on this replica relative to the other replicas in the same availability group. The value is an integer in the range of 0 - 100. These values have the following meanings:

  - 0 indicates that this availability replica will never be chosen for performing backups. This choice is useful for a remote availability replica for which you never want backups to fail over.

  - 1 indicates the lowest priority that an availability replica could be chosen for a backup operation.
    If **BACKUP_PRIORITY = 1**, the availability replica will be chosen only if no higher priority availability replicas are available.

  - 100 indicates the highest priority that an availability replica could be chosen for a backup operation.

Follow the procedures provided in the article "Configure Backup on Availability Replicas (SQL Server)" on the Microsoft MSDN website http://msdn.microsoft.com/en-us/library/hh710053.aspx to decide which replica the backups will be run on.

(i) **Note:** Federated backup operations fail if a replica has the **Readable Secondary** field set to **No**. For backups to succeed, in the **Availability Group Properties** window, set the **Readable Secondary** configuration option to either **Yes** or **Read-intent only** for each of the primary and secondary replicas.

(i) **Note:** The following limitations apply to SQL Server backups of a secondary SQL Server replica in an Availability Group:

- For full backups of a secondary replica, SQL Server supports only copy-only backups.

- Backup promotion is not supported for copy-only backups, including copy-only transaction log backups.

- SQL Server does not support differential backups on secondary SQL Server replicas.

Specifying a retention policy for manual backups

A retention policy defines the length of time that a backup or clone save set is retained and browsable.

(i) **Note:** NetWorker 8.2.x uses separate policies for browse and retention. Refer to the *NetWorker Module for Microsoft for SQL VDI User Guide* version 8.2 SP1 for information on the browse and retention policies for NetWorker 8.2.x.

If you specify a retention policy with a manual backup from the command prompt, the retention policy takes effect for all the save sets in the manual backup. To specify a retention policy with a manual backup at the command prompt, type the following command:

```
nsrsqlsv -y
```

The retention policy value must be typed in time and date formats accepted by the `nsr_getdate` program.

The *NetWorker Command Reference Guide* or the UNIX man pages provide more information about the `nsr_getdate` program. Refer to the "Command Prompt" chapter for information about using the `nsrsqlsv` command for manual backups.

If you do not specify a retention policy for a manual backup, the save sets in a manual backup adopt the browse policy of the client resource. If multiple client resources exist for the NetWorker host, the client resource with the longest retention policy is adopted. However, if a retention policy is set up for the media pool to which the backup is directed, the retention policy is the longer of either:

- The client resource retention policy
- The media pool retention policy

The *NetWorker Administration Guide* provides more information about retention policies.

Performing manual backups by using the NMM SSMS plug-in GUI

It is recommended that you use the NMM SQL Server Management Studio (SSMS) plug-in GUI to perform SQL Server VDI manual backup operations.

About this task

In the **NetWorker** window, the **Script** view is available in each page. Click **Script** to generate a command prompt equivalent script, which you can use to create a `.bat` file to perform scheduled backups, automation, and other tasks.

Procedure

1. In the SQL Server Management Studio, select the SQL Server instance that you want to manually back up and open the **NetWorker** window.

2. In the **NetWorker** window, on the **Backup** tab, under **Select a page**, click **General**.

   The **General** page appears.
   The following figure shows the **NetWorker** window open to the **Backup** tab, on the **General** page.

Figure 10 NetWorker Backup General page

3. Under **Source**, make the following selections:

    a. In the **SQL Server Instance** field, select the SQL Server instance that you want to back up.

    b. In the **Database Filter** list, filter the databases by selecting one of the following:

    *   Select **Non AAG Databases** to show all databases that are not part of an Always On Availability Group (AG).

    *   Select the name of an AG to show all databases that are within that AG.

    The list is populated with databases available for backup. If an AG is selected from the list, under **Connections**, the **Cluster** and the **Backup Preference** appear.

    c. Select one or more databases that you want to back up.

    d. Under **Client type**, perform one of the following actions for availability groups:

    *   To use the cluster name, select **Cluster**.

    *   To use an availability group listener, select **Listener**, and then select a listener from the list.

    (i) Note: The **Client type** setting can be modified only for availability group backups. In non-availability group environments, it is greyed out.

    e. In the **Backup type** list, select the backup type.

(i) **Note:** The backup levels are the same regardless of the version of the NetWorker server you are using.

   f. (Optional) To perform a copy-only backup, select **Copy-only backup**.

   A copy-only backup backs up the SQL server without interrupting the existing backup schedule or log chain.

4. Under **Destination**, make the following selections:

   a. In the **NetWorker Server** list, select the destination NetWorker Server for the backup.

   You can select a NetWorker server from the available list. Click **Update** to force an update of the list of available NetWorker servers.

   b. In the **Debug level** box, select the level of troubleshooting information to be sent to the backup status window during the backup operation.

   Levels range 0–9, with 0 representing no information and 9 representing the most amount of information.

5. (Optional) To specify additional backup options, under **Select a page**, click **Options**.

The **Options** page appears.
The following figure shows the **NetWorker** window open to the **Backup** tab, on the **Options** page.

Figure 11 NetWorker Backup Options page
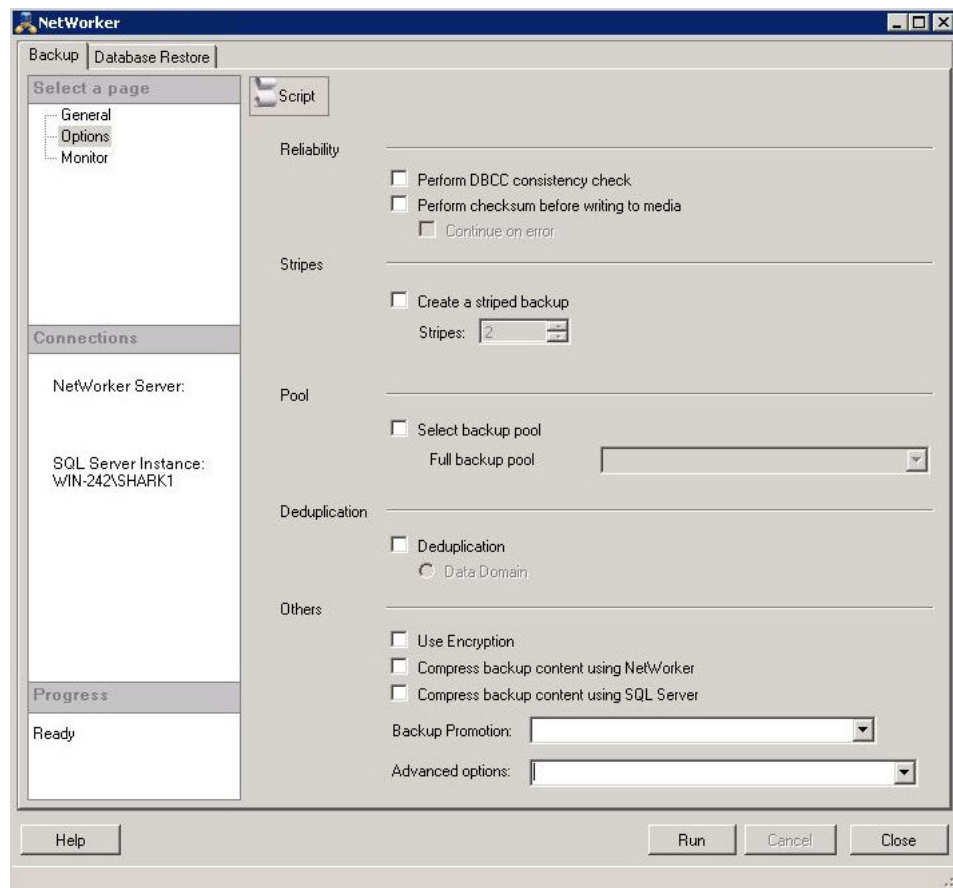
6. (Optional) Under **Reliability**, select the following options as required:

   • To perform a thorough check of the entire database before the backup is performed, select **Perform DBCC consistency check**.

> (i) Note: When **Perform DBCC consistency check** is selected in the SSMS plug-in, the script generated contains -j ALL even when specific databases are selected. This does not affect functionality.

- To perform a checksum operation with the backup and save the information to the backup media, select **Perform checksum before writing to media**. Another checksum is performed before a restore to ensure that it matches the backup.
  A checksum detects a partial backup or restore state. The SQL Server verifies the checksum by calculating a local result and comparing the local value with the stored value.

- To continue with backup and restore even if errors are discovered during the checksum, select **Continue on error**.

7. (Optional) Under **Stripes**, select the following options as required:

   a. To enable the SQL Striped feature, select **Create a striped backup**.

   b. In the **Stripes** box, type or select the number of stripes to use during the backup.

   NMM supports a maximum of 32 stripes. However, the number of stripes cannot exceed the value set for NetWorker client parallelism.

   The section Striped Backup and Recovery provides more information about striping.

8. (Optional) Under **Pool**, select the following options as required:

   a. To select the pool where the backup is stored, select **Select backup pool**.

   b. From the **Full backup pool** list, select the pool that you want to back up to.

   The pools that are listed are populated from the NetWorker server.

9. (Optional) Under **Deduplication**, select the following options as required:

   a. To enable deduplication for the backup, select **Deduplication**.

   b. To use a Data Domain device to store the backup, select **Data Domain**.

10. (Optional) Under **Others**, select the following options as required:

    > (i) Note: It is recommended that no form of encryption or compression be used with Data Domain because these reduce the deduplication ratio.

- To back up the data with AES encryption, select **Use Encryption**.
  Data is encrypted with the default or current pass phrase which is provided by the NetWorker Server. If the NetWorker Server has a different pass phrase during restore, you must specify the pass phrase that was used at the time of backup. The *NetWorker Administration Guide* provides complete information about AES encryption, and setting the pass phrase.

- To apply XBSA compression to all marked databases before the backup data is written to the storage device, select **Compress the backup content using NetWorker**.
  In the same manual backup, you cannot back up some databases with compression and others without.

  Compressing data for a backup generates less network traffic and uses less backup media space, but it consumes additional CPU resources. Most tape devices perform compression, which makes software compression unnecessary.

- To compress the backup using SQL Server, select **Compress the backup content using SQL Server**.
  The Microsoft SQL Server product documentation provides more information.

- To specify backup promotion options, select one of the following values from the **Backup Promotion** list:

- ■ `ALL`: Enables backup promotion to occur in any applicable scenario.
- ■ `NONE`: Disables backup promotion. Logs a warning when backup promotion would normally occur.
- ■ `NONE_WITH_WARNINGS`: Disables backup promotion. Logs a warning when backup promotion would normally occur.
- ■ `SKIP_RECOVERY_MODEL`: Disables database restore model change detection. Backup promotion as a result of restore model change will not occur, but backup promotion in other scenarios will still occur.

  (i) Note: If you leave the **Backup Promotion** field on the client side blank and the client resource for the server has backup promotion set to `NONE`, then backup promotion will not occur. However, if you select an option from the **Backup Promotion** list on the client side, the option will override the server setting.

- • To apply advanced options to the backup operation, in the **Advanced options** field, type or select advanced options. Separate multiple entries with a comma, for example: `BUFFERCOUNT=2, READ_WRITE_FILEGROUPS`

  NMM supports the following advanced backup options:

  - ■ `BUFFERCOUNT=number_of_IO_buffers`: Specifies the total number of IO buffers that can be used during the backup operation.
  - ■ `READ_WRITE_FILEGROUPS`: Backs up only the read/write (active) filegroups within the database.

11. To start the backup, click **Run**.

Results

The backup operation runs.
To view the status of the backup operation, open the **Monitor** page.

The following figure shows the **NetWorker** window open to the **Backup** tab, on the **Monitor** page.

Figure 12 NetWorker Backup Monitor page
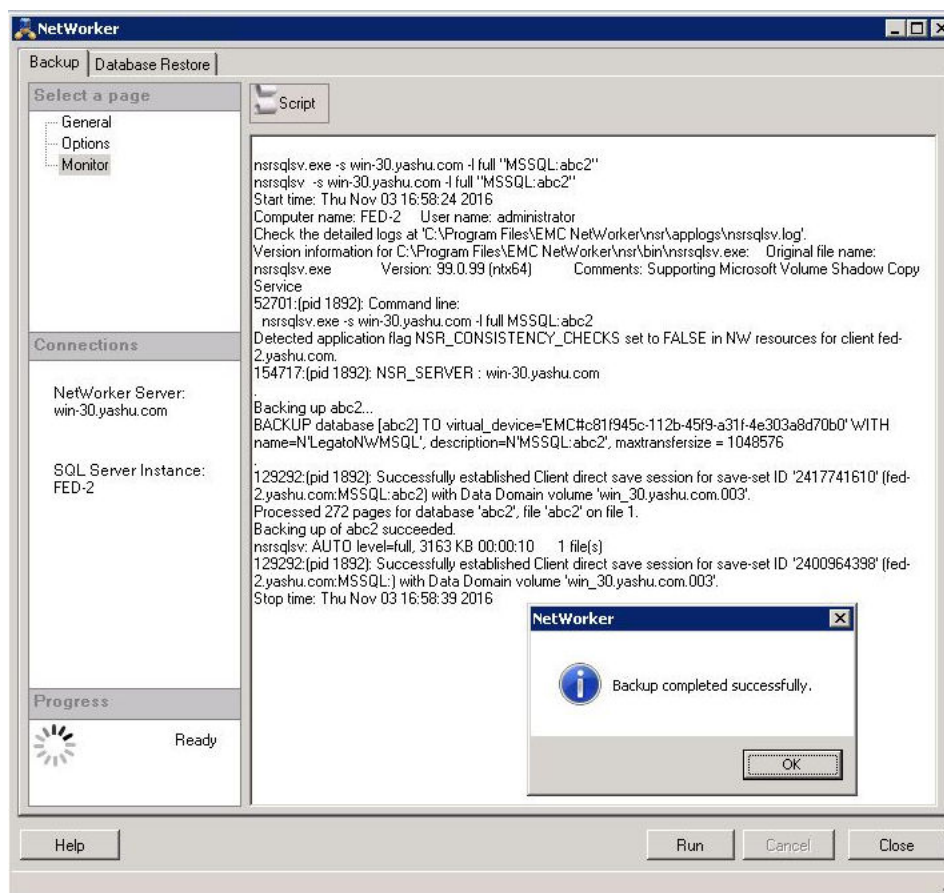
Performing manual backups from the command prompt

The `nsrsqlsv` command is used to back up SQL Server data objects, which consist of files, file groups, and databases.

About this task

To start a backup, specify the `nsrsqlsv` command and its options from a Windows command prompt.

Before you use the backup command, review the following considerations:

- Ensure that each command option is either lowercase or uppercase and, frequently both the cases of a letter are in the set of command options. Case is very important when specifying command-line flags.
  For example, `-c` specifies the NetWorker client name, while `-C` copies the database that is being restored to either the same SQL Server or a different SQL Server.

- Ensure that you use correct spacing. Depending on the command option, the space separator between an option and its corresponding argument can be optional, required, or not allowed. For example, both of the following expressions are valid:

  `-l` *`backup_level`*

  `-l`*`backup_level`*

  The following expression is invalid because a space is not allowed between the + argument and *log_mark_name*:

```
-M + log_mark_name
```

- Ensure that you use brackets ([ ]) to denote the optional portions of a command. When starting an actual backup or recovery operation, do not include brackets.
- Ensure that data items follow all other command options and parameters at the command prompt.

Command syntax for nsrsqlsv

The nsrsqlsv command syntax is:

nsrsqlsv **[-CGjqRTvkuHXZ]**

[-a "option_key=value"]

[-A virtual-server]

[-b pool]

[-c client]

[-C]

[-D debug_level]

[-f aes]

[-g group]

[-G]

[-h "database_name"]

[-H]

[-I text_file]

[-k]

[-j database names]

[-l backup_level]

[-m masquerade]

[-N name]

[-O "Option1, Option2, Option3"]

[-P password]

[-q]

[-R]

[-s networker_server]

[-S number_of_stripes]

[-T]

[-u]

[-U username]

[-v]

[-w browse]

[-X]

[-y retention_period]

[-z federated backup options]

```
[-Z]
```

Command options for nsrsqlsv

The following table outlines the list of options available for the `nsrsqlsv` command.

Table 15 Command options for `nsrsqlsv`

Command options	Descriptions
`-a` `"NSR_INCLUDE_AAG_DATABASE=None"`	Excludes Always On Availability Group databases from instance-level backups.
`-a` `"SKIP_SIMPLE_DATABASE={TRUE | FALSE}"`	Specifies whether to exclude simple restore model databases from a logs only backup. The default value is FALSE. This option is useful when SIMPLE restore model databases are part of a logs only backup. SIMPLE restore model databases do not support logs only backups and the backups are promoted to level full by default. However, this option does not affect the backup of system databases, such as master and msdb, which are promoted to level full during a logs only backup and cannot be omitted from the backup. This option does not apply to full and cumulative incremental backups.
`-a` `"NSR_SKIP_NON_BACKUPABLE_STATE_DB={TRUE | FALSE}"`	Specifies whether to check the status of the selected databases and ignore the databases that are not ready or available to be backed up. If the status of a database is online, the database is ready and available to be backed up. If the status of a database is offline, emergency (single user mode), suspect, restoring, recovering, or recovery pending, the database is not ready or available to be backed up. The default value is FALSE.
`-a "NSR_COPY_ONLY={TRUE | FALSE}"`	Performs a copy-only backup from the SQL Server. Copy-only backups do not disturb the SQL Server backup chain and do not affect backup-level promotion and log truncation. This option works only if you specify either full or transaction log as the backup type in the `-l <Backup_Type>` option. The default value is FALSE. You rarely need to perform copy-only transaction log backups because they are generally required only to perform online restores.
`-a` `"SKIP_SYSTEM_DATABASE={TRUE | FALSE}"`	Specifies whether to skip the system databases during the instance-level transaction log backups. The default value is False.
`-a "BACKUP_PROMOTION={ALL|NONE|NONE_WITH_WARNINGS|SKIP_RECOVERY_MODEL}"`	Specifies backup promotion options. The following values are valid: • `ALL` (Default): Enables backup promotion to occur in any applicable scenario. • `NONE`: Disables backup promotion. Logs a warning when backup promotion would normally occur. • `NONE_WITH_WARNINGS`: Disables backup promotion. Logs a warning when backup promotion would normally occur. • `SKIP_RECOVERY_MODEL`: Disables database recovery model change detection. Backup promotion as a result of recovery model change will not occur, but backup promotion in other scenarios will still occur.

Command options	Descriptions
	The setting specified with this option applies to every database in the SQL instance when this setting is specified along with the SQL instance level save set **MSSQL:** or **MSSQL$**_<instance name>_.
	Similarly, the setting specified with this option applies only to specified databases when this setting is specified with a database level save set **MSSQL:** _<database name>_ or **MSSQL$**_<instance name>_: _<database name>_.
`-a "Deduplication backup=yes"`	Enables backup deduplication.
`-a "Device interface=`_{data domain | avamar}_`"`	Specifies whether data deduplication should be set with Data Domain or Avamar.
`-a "Deduplication node=`_node_name_`"`	Specifies the deduplication node to use for the backup.
`-a "Deduplication client name=`_client_`"`	Specifies the deduplication client to use for the backup.
`-A` _virtual_server_	Specifies the virtual server name when a SQL Server is configured to run in an MSCS cluster.
`-b` _pool_name_	Assigns a destination pool for a save set. Specifying `-b` _pool_name_ overrides all other pool-selection criteria that is provided by the NetWorker client. The pool must be created with a corresponding label template before a command that includes the `-b` option runs.
`-c` _client_	Specifies the NetWorker client name for which the SQL Server data is to be backed up.
`-C`	Specifies compression of the backup data before the NetWorker client sends the data to the NetWorker server.
`-D <debug_level>`	Generates detailed logs that you can use to troubleshoot backup issues. The default value is 0 (zero). If you want to enable debug logs, specify a level between 1 and 9.
`-f aes`	Enables the NetWorker server to back up data using AES encryption.
`-g` _group_	Specifies the savegroup. The NetWorker server and the `savegrp` command use the group parameter to select the media pool.
`-G`	Specifies a NO_LOG transaction log backup before backing up the database. This command option is valid only for level full backups. This option is deprecated in SQL Server 2008 or later.
`-h <`_"db_name_with_optional_wildcard"_`>`	Excludes databases from only the client-initiated backups. ⓘ **Note:** Do not use this option for the server-initiated backups. You can use this option to specify exact database names or use wildcard characters. Two wildcard characters are supported: • Question mark (?): Matches any single character • Asterisk (*): Matches zero to unlimited characters

<div align="center">**Table 15** Command options for `nsrsqlsv` (continued)</div>

Command options	Descriptions

	When you use wildcards you must enclose the database name in square brackets, for example, *["DB?"]*.
	
	For example, consider the following scenarios:
	
	• To exclude only DB_1 and DB_2 from the backup, type the following command:
	`nsrsqlsv -s bv-customer.belred.emc.com -h "DB_1" -h "DB_2" MSSQL:`
	
	• To exclude all databases that are named with the format of DB_*x*, such as DB_9 and DB_a, type the following command:
	`nsrsqlsv -s bv-customer.belred.emc.com -h ["DB_?"] MSSQL:`
	
	• To exclude all databases with names ending in DB, type the following command:
	`nsrsqlsv -s bv-customer.belred.emc.com -h ["*DB"] MSSQL:`

`-H`	Specifies the NORECOVERY option when backing up transaction logs. This option leaves the database in Restoring state.
`-I text_file`	Specifies a text file that lists multiple SQL Server save sets. The `-I` option can also be specified with the `nsrsqlsv` command for the **Backup Command** field in the NetWorker Administration GUI.
`-k`	Performs checksum before writing to media.
`-j database_name`	Performs a database consistency check (DBCC) before the backup is performed. For example, `-j testdb3,CopyOfDB010,test4` runs a consistency check for databases testdb3, CopyOfDB010, and test4. ⓘ **Note:** When **Perform DBCC consistency check** is selected in the SSMS plug-in, the script generated contains `-j ALL` even when specific databases are selected. This does not affect functionality.
`-l backup_level`	Specifies the backup level. The following values are valid: • full • diff • txnlog The "Combining data objects to create backup levels" section provides more information about which backup levels are supported for various SQL Server data objects. The `-l` option is valid only for manual backups that are initiated from a Windows command prompt on a client host. For scheduled backups, set the backup level in the **Set Level** dialog box of the schedule resource in the NetWorker Administration GUI. Do not use the `-l` option when you start a backup in the NetWorker Administration GUI. Sample outputs for each of the three different levels of backups follow: • Full backup of the database to a specified NetWorker server:

Command options	Descriptions

	nsrsqlsv -s *server* -c *client* -b *poolname* -l **full** **MSSQL:** *dbname*
	
	nsrsqlsv -s swraj -c SQL2012RC1Named -b Sub9VDI -l full MSSQL:5
	
	`43708:(pid 6004):Start time: Sat Jan 28 09:07:42 2012`
	`43621:(pid 6004):Computer Name: SQL2012-NODE3 User Name:`
	`Administrator`
	`NSR_BACKUP_LEVEL: full;`
	`NSR_CLIENT: SQL2012RC1Named.joy.com;`
	`NSR_DATA_VOLUME_POOL: Sub9VDI;`
	`NSR_LOG_VOLUME_POOL: Sub9VDI;`
	`NSR_SAVESET_NAME: "MSSQL:5";`
	`NSR_SERVER: Swraj.joy.com;`
	`37994:(pid 6004):Backing up 5...`
	`4690:(pid 6004):BACKUP database [5] TO`
	`virtual_device='EMC#4018d580-f511-4457-abc3-a62c4c3f0ff9'`
	`WITH name=N'EMCNWMSQL'`
	`53085:(pid 6004):Backing up of 5 succeeded.`
	`nsrsqlsv: MSSQL: 5 level=full, 2261 KB 00:00:02 1 file(s)`
	`43709:(pid 6004):Stop time: Sat Jan 28 09:07:49 2012`
	
	• Cumulative incremental backup:
	nsrsqlsv -s *NetWorker_server_name* -l **diff** **MSSQL:** *dbname*
	
	Output similar to the following appears for a backup of a database named my_database:
	
	`nsrsqlsv: Backing up my_database...`
	`nsrsqlsv: BACKUP database my_database TO`
	`virtual_device='BSMSQL' WITH differential, stats`
	`nsrsqlsv: my_database level=diff, 719 KB 00:00:05 1 file(s)`
	
	• Logs-only backup:
	nsrsqlsv -s *NetWorker_server_name* -l **txnlog** **MSSQL:** *dbname*
	
	Output similar to the following appears for a backup of a database named my_database:
	
	`nsrsqlsv: Backing up my_database...`
	`nsrsqlsv: BACKUP log my_database TO virtual_device`
	`='BSMSQL'`
	`nsrsqlsv: my_database level=txnlog, 61 MB 00:00:05 1`
	`file(s)`
	
	At least one SQL Server data item (file, file group, or database) must be specified, and the data items and list of data objects must follow all other command options and parameters at the command prompt.

-m *name*	Specifies the cluster or availability group listener name.
-N *name*	Specifies the symbolic name of the save set. By default, the most common prefix of the path arguments is used as the save set name. When performing a federated backup of an availability group, you must specify the save set name with **-N**.
-O *"option1, option2"*	Specifies advanced backup options. You can specify the following advanced backup options:

Table 15 Command options for `nsrsqlsv` (continued)

Command options	Descriptions

	• **BUFFERCOUNT=**_number_of_IO_buffers_
	This option specifies the total number of IO buffers that can be used during a backup operation.
	• **READ_WRITE_FILEGROUPS**
	This option backs up only the read/write (active) filegroups within the database.
	If you are specifying multiple options, separate each argument with a comma. For example:
	`nsrsqlsv -s NetWorker_server_name -O`
	`"BUFFERCOUNT=number_of_IO_buffers,`
	`READ_WRITE_FILEGROUPS" MSSQL:`

-P _password_	Specifies the SQL Server user password. When the **-U** user name command option is specified, the password command option must also be provided, as follows:
	`nsrsqlsv -s NetWorker_server_name -U username -P`
	`password MSSQL:`
	Use the SQL Server username and password to log in to SQL Server with SQL server authentication.

-q	Displays `nsrsqlsv` messages in quiet mode; only summary information and error messages appear.
-R	Specifies the NO_TRUNCATE option when backing up transaction logs.
-s _NetWorker_server_name_	Specifies the name of the NetWorker server that will be used for the backup operation.
-S _number_of_stripes_	Backs up the specified data items using stripes. To use backup and restore striping successfully, see the striping configuration described in Striped Backup and Recovery.
-T	Performs a TRUNCATE_ONLY transaction log backup before backing up the database and is valid for only full backups. ⓘ Note: This flag is deprecated in SQL Server 2008 or later.
-u	Continues the backup even in the event of a checksum error.

-U _username_	Specifies the SQL Server username. When this command option is specified, the **-P** password command option must also be provided, as follows:
	`nsrsqlsv -s NetWorker_server_name -U username-P password`
	`MSSQL:`
	Use the SQL Server username and password to log in to SQL Server with SQL server authentication.

-v	Displays the `nsrsqlsv` command messages in verbose mode, providing detailed information about the progress of the backup operation.
-X	Indicates that SQL Server internal backup compression is used.

Table 15 Command options for `nsrsqlsv` (continued)

Command options	Descriptions
`-y retention_period`	Specifies the retention period of the backup. The value must be typed in time and date formats accepted by the `nsr_getdate` program. The *NetWorker Command Reference Guide* or the UNIX man pages provide more information about the `nsr_getdate` program.
`-z`	Applies to the backup of up databases for a SQL Server 2005, and is usually used in the online restore scenario from the command prompt. The `-z` option specifies that the logs-only (transaction log) backup after restore is not promoted to full backup. Without the `-z` option, the backup is promoted to full.
`-z federated_backup_option`	Specifies federated backup options. Use the following values with the `-z` option: • `-z FEDERATED_SLAVE=true` This option denotes that the process is SQL Federated secondary process. • `-z FEDINDEX_NAME =SQL instance name under which the backups are browsable` For example, `MSSQL` or `MSSQL$InstanceName` • `-z FEDCLIENT_NAME =Windows cluster client name` This option is useful for cluster federated backups. When you use the `-z` option, you must also use the `-N` option to specify the save set name. For example, to perform an incremental federated backup, type the following command: ``` nsrsqlsv.exe -s mars.jupiter.com -m WIN-FED-3 -a device interface=device interface -a device interface data domain - a Data Domain interface=IP -b Default -l incr -q -z FEDERATED_SLAVE=true -z FEDINDEX_NAME=MSSQL -z FEDCLIENT_NAME=cluster_fed -N MSSQL#MARS:MARSDB MSSQL:MARSDB ```

Backup and recovery command syntax for SQL Server data

Use the additional command syntax that is shown in the following table to back up or restore SQL Server data with the nsrsqlsv and nsrsqlrc backup and recovery commands.

You can specify more than one data object and can combine different types of data. You must specify the SQL data objects with the syntax shown in the following table.

Table 16 Command syntax for SQL Server data

SQL Server data	Syntax for SQL Server data objects
All databases in the SQL Server storage hierarchy (optional)	`MSSQL:` Typing MSSQL: yields an instance-level backup of all databases on the SQL Server host.

Table 16 Command syntax for SQL Server data (continued)

SQL Server data	Syntax for SQL Server data objects

Specified databases	`MSSQL:dbName`
	
	or
	
	`[MSSQL:dbName MSSQL:dbName2 ...]`

All file groups in specified databases	`MSSQL:dbName`
	
	or
	
	`[MSSQL:dbName. MSSQL:dbName2 ...]`

Specified file groups in specified database	`MSSQL:dbName.fgName`
	
	or
	
	`[MSSQL:dbName.fgName MSSQL:dbName.fgName2`
	`MSSQL:dbName2.fgName MSSQL:dbName2.fgName2 ...]`

Specified files in specified file groups in specified databases	`MSSQL:dbName.fgName.filename`
	
	or
	
	`[MSSQL:dbName.fgName.filename`
	`MSSQL:dbName.fgName2.filename`
	`MSSQL:dbName2.fgName.filename`
	`MSSQL:dbName2.fgName2.filename ...]`

Specifying `MSSQL` before each data object name is optional and does not affect the expression or the resulting operation. However, when `MSSQL` is specified, it must be followed by a colon (:).

For example, the following two commands are equivalent:

```
nsrsqlsv -s NetWorker_server_name dbName.fgName
nsrsqlsv -s NetWorker_server_name MSSQL:dbName.fgName
```

In a non-clustered, named instance configuration, `MSSQL$` is required, followed by the instance name and a colon. For example:

```
nsrsqlsv -s NetWorker_server_name MSSQL$instanceName:dbName.fgName
```

Syntax for a named instance configuration

When a configuration contains non-clustered named instances of the SQL Server, you must specify the name of the instance before the data.

```
MSSQL$instanceName:[dbName ...] [.fgName ...] [.fileName ...]
```

(i) Note: The `nsrsqlsv` and `nsrsqlrc` commands support specification of only a single instance. If save sets for more than one instance are specified, the backup fails. The `nsrsqlrc` command supports mixing instances for a copy restore operation.

Example 3 Back up all databases in an instance

To back up all the databases for instanceOne, type the following:

```
nsrsqlsv -s NetWorker_server_name MSSQL$instanceOne:
```

Example 4 Recover several file groups in an instance

To recover several file groups for instanceTwo, type the following:

```
nsrsqlrc -s NetWorker_server_name MSSQL$instanceTwo:dbName.fgName
MSSQL$instanceTwo:dbName.fgName2
```

Example 5 Using clustered instance SQL Server virtual server names with -A or -c

Instead of using clustered named instances in this syntax, use clustered instance SQL Server virtual server names with -A or -c options, as shown in the following command:

```
nsrsqlsv -s NetWorker_server_name -A SQL_virtual_server_DNS_name -c
SQL_virtual_server_DNS_name MSSQL:
```

where:

- *NetWorker_server_name* is the hostname of the NetWorker server.
- *SQL_virtual_server_DNS_name* is the Domain Name System (DNS) name for the SQL Server virtual server.

Create a client resource under the SQL Server virtual server DNS name.

For scheduled backups of a SQL Server virtual server client, you do not need to specify the -A or -c option with the SQL Server virtual server name. The savegrp process automatically specifies the virtual server name to the nsrsqlsv process by using the -m option.

Syntax for names containing a period, backslash, or colon

NMM provides command line syntax that enables you to back up and recover file names, file groups, and databases containing a period (.), backslash (\), or colon (:). By typing a backslash before the period or backslash, the nsrsqlsv and nsrsqlrc commands interpret the period or backslash as a literal character.

The tables in this section list the syntax for file names, file groups, and databases containing a period, backslash, colon, or any combination of the three.

The following notes apply to the information in the tables:

- The syntax that is shown in the right column applies to both the nsrsqlsv and nsrsqlrc commands.
- The notation MSSQL: is optional only for the nsrsqlsv command.
- A single period (.) continues to delimit SQL identifiers.
- The syntax also applies to named instances.
- The backslash period (\.) character sequence replaces each literal period in the SQL identifier.
- The double backslash (\\) character sequence replaces each literal backslash in the SQL identifier.

The following table lists the syntax for file names, file groups, and databases containing a period.

Table 17 Command syntax for names containing a period

Name visible from SQL utilities	Equivalent command-line syntax
SQL database named MyDatabase.COM	`MSSQL:MyDatabase\.COM` `MSSQL$MyInstance:MyDatabase\.COM`

SQL file group named MyFileGroup.2 for the SQL database named MyDatabase.COM	`MyDatabase\.COM.MyFileGroup\.2`
	`MSSQL:MyDatabase\.COM.MyFileGroup\.2`
	
	`MSSQL$MyInstance:MyDatabase`
	`\.COM.MyFileGroup\.2`

SQL file named MyFile.2, which is a member of the SQL file group named MyFileGroup.2 for the SQL database named MyDatabase.COM	`MyDatabase\.COM.MyFileGroup\.2.MyFile\.2`
	`MSSQL:MyDatabase\.COM.MyFileGroup`
	`\.2.MyFile\.2`
	
	`MSSQL$MyInstance:MyDatabase\`
	`\COM.MyFileGroup\\2.MyFile\\2`

The following table lists the syntax for file names, file groups, and databases containing a backslash.

Table 18 Command syntax for names containing a backslash

Name visible from SQL utilities	Equivalent command-line syntax

The SQL database named MyDatabase\COM	`MyDatabase\\COM`
	`MSSQL:MyDatabase\\COM`
	
	`MSSQL$MyInstance:MyDatabase\\COM`

The SQL file group named MyFileGroup\2 for the SQL database named MyDatabase\COM	`MyDatabase\\COM.MyFileGroup\\2`
	`MSSQL$MyInstance:MyDatabase\`
	`\COM.MyFileGroup\\2`

The following table lists the syntax for file names, file groups, and databases containing a colon.

Table 19 Command syntax for names containing a colon

Name visible from SQL utilities	Equivalent command-line syntax

SQL database named MyDatabase:COM	`MyDatabase:COM`
	`MSSQL:MyDatabase:COM`
	
	`MSSQL$MyInstance:MyDatabase:COM`

SQL file group named MyFileGroup:2 for the SQL database named MyDatabase:COM	`MyDatabase:COM.MyFileGroup:2`
	`MSSQL:MyDatabase:COM.MyFileGroup:2`
	
	`MSSQL`
	`$MyInstance:MyDatabase:COM.MyFileGroup:2`

SQL file named MyFile, which is a member of the SQL file group named MyFileGroup:2 for the SQL database named MyDatabase:COM	`MSSQL`
	`$MyInstance:Mydatabase:com.MyFileGroup:2.`
	`MyFile`

Example backup command syntax

Back up a WSFC cluster

The Windows account that you use to back up the cluster must have WSFC administrator privileges. To determine which accounts have WSFC administrator privileges, refer to the WSFC online help. If the Windows account does not have WSFC administrator privileges, NMM cannot communicate with WSFC and the various WSFC cluster resources, including the SQL Server virtual servers.

To back up a SQL virtual server database, type the following:

`nsrsqlsv -A *SQL_virtual_server_name* -c *SQL_virtual_server_name* -s *NetWorker_server_name* MSSQL:*dbName*`

where:

- *SQL_virtual_server_name* is the virtual server name when the SQL Server is configured to run in a WSFC cluster.
- *NetWorker_server_name* is the NetWorker server that is designated for the backup.
- *dbName* is the name of the SQL Server database that is to be backed up.

Specifying `-A *SQL_virtual_server_name*` starts the following actions:

- Contacts the SQL virtual server.
- Creates save set entries under *SQL_virtual_server_name* in the NetWorker client index.

Perform a full backup of an Always On Availability Group

`nsrsqlsv.exe -s nwsrvr -m automation -a device interface="data domain" -b Default "MSSQL$SQ12INST4#sql2012-aag3:"`

where:

- `nwsrvr` is the NetWorker server
- `automation` is the Windows cluster name
- `MSSQL$SQ12INST4#sql2012-aag3:` is the save set name:
  - `MSSQL` is a mandatory term.
  - `SQ12INST4` is the SQL Server instance name.
    If you use a named instance, ensure that you perform the backup by using the same instance name, even if the SQL Server has failed over to another node in the Always On Availability Group.
  - `#` indicates a federated backup.
  - `sql2012-aag3` is the Always On Availability Group name.
- `MSSQL$SQ12INST4#sql2012-aag3:` is the backup object name, where all the databases of the sql2012-aag3 Always On Availability Group are backed up.

Back up a specific database in an Always On availability group

To restrict the backup to specific databases, specify the database names. For example, to back up database1 only, type `"MSSQL$SQ12INST4#sql2012- aag3:database1" "MSSQL$SQ12INST4#sql2012-aag3:database1"`

Back up multiple instances on multiple nodes

When you have multiple instances on multiple nodes in an Always On availability group, select one instance name to use as the save set name and target for the Always On availability group backup.

Use that same instance name when naming the backup object, regardless of which node in the availability group the backup is started on.

For example, consider the following scenario:

- There are two instances, SQ12INST4 and SQ12INST5, are present on Node 1 and Node 2 respectively.

- You want to use SQ12INST4 for the backup.

- Node 1 is down, and as the result, the SQ12INST4 instance is down.

- You must perform the backup on Node 2 because Node 1 is down.

In this scenario, to use SQ12INST4, you must specify the save set and the backup object as `"MSSQL$SQ12INST4#sql2012-aag3:"` `"MSSQL$SQ12INST4#sql2012- aag3:"` instead of `"MSSQL $SQ12INST5#sql2012-aag3:"` `"MSSQL $SQ12INST5#sql2012-aag3:"`.

CHAPTER 5

Scheduled Backups

This chapter includes the following sections:

Overview of scheduled backup

The most reliable way to protect SQL Server data is to schedule backups of the SQL Server at regular intervals. Scheduled backups ensure that all SQL Server data is automatically saved.

Scheduled backups can be set for backup levels full, logs only, or cumulative incremental, and they can be configured to run at any time.

Review the *NetWorker Administration Guide* before reviewing the information in this chapter.

Prerequisites

Before you perform scheduled database backups with a SQL virtual server, you must meet several prerequisites.

Ensure that the following prerequisites are met before you perform scheduled backups:

- The NetWorker interface displays diagnostic attributes. In the **Administration** window, click **View** > **Diagnostic Mode**.
- When backing up filestream data, in the SQL Server Configuration Manager, on the **Properties** page of the SQL Server instance, select **Allow Remote Clients Access to Filestream Data**. This action is a requirement for backups to be successful. The SQL Server documentation provides details about the SQL Server Configuration Manager.

Prerequisites for cluster environments

You must meet additional prerequisites for cluster environments:

- Create a NetWorker client that is configured for database backups for each SQL virtual server that you want to protect.
- Configure the NetWorker client to either:
  - Back up all databases with `MSSQL:`.
  - Back up specific databases with **`MSSQL:User_Database`**.
    Add more databases to an existing NetWorker client or create a new database client when additional databases are added to an existing NetWorker client or a new database client is created.
- Specify virtual servers by typing the following command at the command prompt:
  `-A SQL_virtual_server`

  Use the fully qualified domain name for the client configuration and the `-A` option to specify the SQL virtual server, as in the following example:

  ```
  nsrsqlsv -sservername -ASQL_virtual_server
  ```

Federated backup preferences for Availability Group databases

You can configure the SQL Server backup preference for Availability Group databases by using either the Microsoft SSMS GUI or the Transact-SQL command.

NMM supports federated backups, during which NMM detects the SQL Server preferred backup setting for the Availability Group and performs the backup at the preferred node. The database administrator can set the backup priority for the Availability Group or a database in the Availability Group and nominate a particular replica for the backup. This feature improves backup performance.

SQL Server 2012 or later provides the following options that determine where backups run for a specific Availability Group replica:

- `AUTOMATED_BACKUP_PREFERENCE`—Specify any one of the following options:

    - `PRIMARY`—Specifies that the backups will always occur on the primary replica. This option is useful if you need backup features, such as creating differential backups, that are not supported when backup operations run on a secondary replica.

    - `SECONDARY_ONLY`—Specifies that backups will never be performed on the primary replica. If the primary replica is the only replica that is online, the backup will not occur.

    - `SECONDARY` (Prefer Secondary)—Specifies that backups should occur on the secondary replica. If the primary replica is the only replica online, the backup is performed using the primary replica. This is the default option.

    - `NONE` (Any replica)—Specifies that you prefer that backup jobs ignore the role of the availability replicas when choosing the replica on which to perform backups. Note backup jobs might evaluate other factors such as backup priority of each availability replica in combination with its operational state and connected state.

- `BACKUP_PRIORITY =n`—Specifies your priority for performing backups on this replica relative to the other replicas in the same availability group. The value is an integer in the range of 0 - 100. These values have the following meanings:

    - 0 indicates that this availability replica will never be chosen for performing backups. This choice is useful for a remote availability replica for which you never want backups to fail over.

    - 1 indicates the lowest priority that an availability replica could be chosen for a backup operation.
    If `BACKUP_PRIORITY = 1`, the availability replica will be chosen only if no higher priority availability replicas are available.

    - 100 indicates the highest priority that an availability replica could be chosen for a backup operation.

Follow the procedures provided in the article "Configure Backup on Availability Replicas (SQL Server)" on the Microsoft MSDN website http://msdn.microsoft.com/en-us/library/hh710053.aspx to decide which replica the backups will be run on.

(i) **Note:** Federated backup operations fail if a replica has the **Readable Secondary** field set to **No**. For backups to succeed, in the **Availability Group Properties** window, set the **Readable Secondary** configuration option to either **Yes** or **Read-intent only** for each of the primary and secondary replicas.

(i) **Note:** The following limitations apply to SQL Server backups of a secondary SQL Server replica in an Availability Group:

- For full backups of a secondary replica, SQL Server supports only copy-only backups.

- Backup promotion is not supported for copy-only backups, including copy-only transaction log backups.

- SQL Server does not support differential backups on secondary SQL Server replicas.

Excluding incompatible databases in backups

If you schedule a backup of a client with the `MSSQL:` save set or named instance save set, and if any of the databases in the protection group are incompatible, the scheduled backup fails.

(i) **Note:** Incompatible databases are silently omitted during manual (non-scheduled) backups that are started from NMM on the client computer.

A database that is in any of the following states causes a scheduled backup to fail because the database is part of a previously configured backup:

- Standby
- Mirror copy
- Recovering
- Restoring
- Recovery Pending
- Suspect
- Offline
- Not recovered
- Loading
- Prerecovery

Definitive results about incompatible databases are available in the `daemon.log` file on the NetWorker server and in the `nsrsqlsv.log` file on the client computer. Examples of error messages listed in the `daemon.log` file after a backup failure:

- `Database 'Acme' cannot be opened because it is offline.`
- `Processing Acme failed, the item will be skipped.`
- `Database 'Acme' is in warm-standby state (set by executing RESTORE WITH STANDBY) and cannot be backed up until the entire load sequence is completed.`
- `Processing Acme failed, the item will be skipped.`

To exclude databases that are not in a state that can be backed up, perform either of the following steps during client resource configuration:

- In the **Client Backup Configuration** wizard on the **Specify the Backups Options** page, select **Skip databases that are in an incompatible state**.
- In the NetWorker Administrator GUI, in the **Client Properties** dialog box, in the **Application Information** field, type `NSR_SKIP_NON_BACKUPABLE_STATE_DB=TRUE`.

Configuring scheduled backups

To configure a client resource, you must set up backup levels, configure a client resource, and then assign data protection policies.

All the procedures that are described in the following table must be performed on a NetWorker server through the NetWorker Administration GUI. Review the *NetWorker Administration Guide* for details.

Table 20 Tasks for configuring SQL Server VDI backups

Tasks	Considerations
Setting up backup levels	The availability of a backup level depends on the type of data that are selected for backup and any SQL Server settings on those objects. You can specify any of the following backup levels: • Full • Logs-only • Cumulative incremental
Configuring a client resource	You can use either of the following methods to configure the client resource: • Client Backup Configuration wizard • **Client Properties** dialog box
Setting data protection policies	To set up the required data protection policies, complete the following tasks: • Create a protection group—The type of group that you create depends on the actions that you plan to perform for the group. • Create a policy—When you create a policy, you specify the name and notification settings for the policy. • Within the policy, create a workflow—When you create a workflow, you specify the name of the workflow, the schedule for running the workflow, notification settings for the workflow, and the protection group to which the workflow applies. • Create one or more actions for the workflow. The *NetWorker Administration Guide* provides more information about data protection policies.

Setting up backup levels

You can specify backup levels other than database full, database cumulative incremental, and database logs only.

The availability of a backup level depends on the type of data that are selected for backup and any SQL Server settings on those objects, as listed in the following table.

Table 21 Backup levels for SQL Server data

SQL Server data objects	Supported SQL Server backup levels		
	Full	Cumulative incremental	Logs only
All databases of SQL default or named instances	Yes	Yes	Yes
Specified databases	Yes	Yes	Yes
All file groups in specified databases	Yes	Yes	N/A
Filestream data in specified databases	Yes	Yes	Yes
Specified file groups in specified database	Yes	Yes	N/A
Specified files in file groups in specified databases	Yes	Yes	N/A

For SQL Server data objects for which logs only backup can be performed, ensure that the SQL Server database options are correctly configured. The Microsoft SQL Server documentation provides more information. Individual items are subject to promotion.

Configuring a client resource

Use NMC to configure each SQL Server host that is to be backed up as a NetWorker client resource. Multiple SQL Server databases that exist on the same SQL Server host can be configured as separate NetWorker client resources.

By default, the system account that is the nsrexecd service account does not have the system administrator (sysadmin) role for the SQL server. Therefore, when you configure a client resource, you must set the username and password of a Windows account that can back up the SQL server. The required privileges are explained in the section Access privileges for backup and recovery on page 44.

You can configure a client resource for scheduled backups in NMC using either of the following tools:

- The Client Backup Configuration wizard
- The **Client Properties** dialog box

Client resources for SQL Server clusters

For federated backups using NMM to succeed, there must be a client resource for the Windows cluster name from which the backup is performed and dummy client resources for the other participating nodes in the cluster.

It is recommended that you use the Client Backup Configuration wizard to configure client resources for cluster environments. The Client Backup Configuration wizard simplifies configuring client resources for SQL Server 2012 or later scheduled backups in a Windows cluster environment by creating dummy client resources for the participating cluster nodes. To use the Client Backup Configuration wizard in a federated environment, ensure that the active node of the Windows cluster contains a SQL Server 2012 or later instance.

If you create the client resources manually using the **Client Properties** dialog box, you must manually create dummy client resources for each participating node in the cluster.

> (i) **NOTICE** NMM supports configuring federated backups of an AG that is configured with multiple SQL server instances, including server instances in a failover cluster. You must use the **Client Properties** dialog box to configure this type of backup.

Configuring a client resource by using the Client Backup Configuration wizard

You must complete the required steps to configure a client resource with the Client Backup Configuration wizard. Click the question mark icon [?] at the lower left of each page of the Client Backup Configuration wizard for details about each field on the page.

About this task

When you configure a client resource for a SQL Server cluster, remember the following points:

- Do not use a short name alias for a virtual server node that is not registered on the NetWorker server with a fully qualified domain client name.

- Select the SQL Server cluster instance or individual databases for backup. Do not select databases from two instances for backup. Databases that are in offline, restoring, or loading state are unavailable for selection.

> (i) **Note:** If you are using NetWorker server 8.2.3 or 8.2.4 and NMM 19.2:
>
> - Configure a regular NetWorker backup group instead of configuring a data protection policy. Do not enable the Snapshot option.
>
> - For the Client Backup Configuration wizard to function correctly, ensure that JRE 8 is installed on the system where NMC is used. While the NMC for NetWorker 8.2.3 and later is compiled with JRE 7, the NMM 19.2 Java plug-in for NMC is compiled with JRE 8.
>
> - The procedure to create a client resource is different between NetWorer server 8.2.3 and NetWorker server 19.2. Follow the procedure provided in the *NetWorker Module for Microsoft for SQL VDI User Guide* version 8.2 SP1 when you are using NetWorker server 8.2.3 or later.

Procedure

1. In the **NetWorker Administration** window, click the **Protection** tab.

2. In the expanded left pane, right-click **Clients** and select **New Client Wizard**.

   The following figure shows the NetWorker **Administration** window menu option to start the Client Configuration wizard.

   Figure 13 Starting the Client Configuration wizard
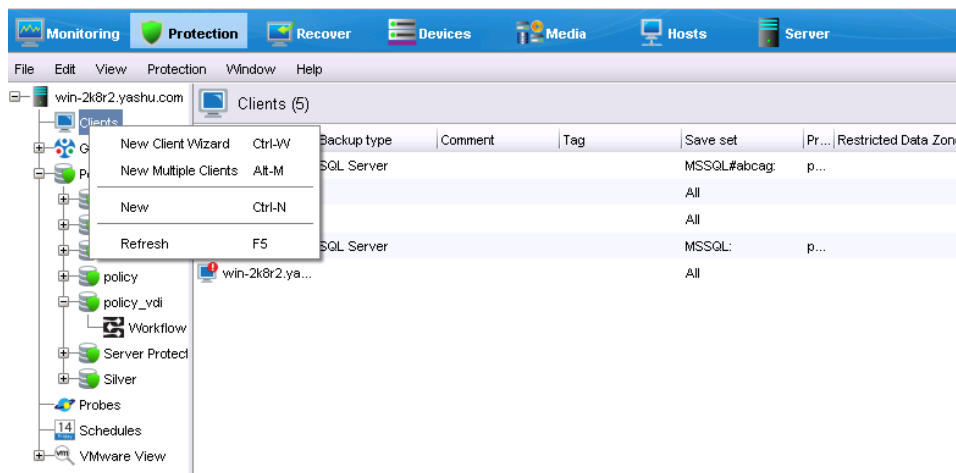
The **Client Backup Configuration** window opens to the **Specify Client Information** page.

3. On the **Specify Client Information** page, specify the following information:

   - In the **Client Name** field, type the hostname or fully qualified domain name (FQDN) of the SQL Server, Windows cluster, or availability group listener.

     (i) **Note:** Do not type the IP address as the client name.

   - In the **Comment** field, type a description for the client resource.

   - In the **Tag** field, type one or more tags to identify this client resource for the creation of dynamic client groups for data protection policies. Dynamic client groups automatically generate a list of clients for a data protection policy that is based on the tags assigned to the client and group.

   - In the **Group** field, select the group that was previously created for the required backup type.

   - In the **Type** area, select **Traditional**.

   The following figure shows the Client Configuration wizard **Specify Client Information** page.

   Figure 14 Specify Client Information page

4. Click **Next**.

5. On the **Specify Backup Configuration Type** page, specify the following information:

   - In the **Client Operating System** field, the client operating system that is used during the configuration setup is automatically displayed.

   - In the **NetWorker Version** field, the NetWorker version that is used during the configuration setup is automatically displayed.

   - From the **Available Application** list, select **SQL Server**.

   - Do not select **Enable NetWorker Snapshot Management on the selected application**.

   The following figure shows the Client Configuration wizard **Specify Backup Configuration Type** page.

Figure 15 Specify Backup Configuration Type page

6. Click **Next**.

7. On the **Select the NetWorker Client Properties** page, specify the following information:

   - In the **Priority** field, select the priority level.

   - In the **Parallelism** field, select the level of parallelism.

   - In the **Remote Access** field, type the required attributes.
     During the client resource configuration, you can control client recover access with the attributes in the **Remote Access** field. The **Remote Access** attribute displays a list of the users that can recover save sets for a client. Depending on the level of security that the files require, add or remove user names.

   - In the **Data Domain Interface** field, select the device type from the list.

   - In the **Block Based Backup** field, leave this option clear. This feature is not supported for SQL Server VDI.

   - The **Client Direct** field, selected by default, enables the client to bypass the storage node and send backup data directly to the storage device.

   (i) Note: To use the default NetWorker Client settings, do not update the options that are provided on the page.

8. Click **Next**.

9. On the **Select the Backup Objects** page, from the prepopulated list, select the SQL Server instance at root level or individual databases for backup.

   (i) Note: Do not select databases from two or separate instances for backup. Also, databases that are in offline, restoring, or loading state are unavailable for selection.

10. Click **Next**.

11. On the **Specify the Virtual Server Credentials** page, in the **Remote User Name** and **Password** fields, type the login information for a Windows account that can backup the SQL server.

    (i) Note: Before performing this step, ensure that the User Account Control for Windows Server 2008 R2 is disabled for administrators.Access privileges for backup and recovery on page 44 provides details.

12. Click **Next**.

    The **Specify the Backup Options** page appears.

13. (Optional) On the **Specify the Backup Options** page, select the following optional settings as required:

    • Select **NetWorker compression** to use NetWorker compression during the backup.

    • Select **256-bit AES software encryption** to use 256-bit AES software encryption.

    • Select **SQL server compression** if you are using SQL Server 2008 or later.

    • Select **Perform checksum before writing to media** to perform a checksum operation with the backup and save the information to the backup media, and then, if required, select **Continue to backup even on checksum error**.

    • Select **Do not truncate log contents when backing up transaction logs** when you are backing up transaction logs for a SQL server 2005.

    • Select **Skip simple databases during incremental backup** to omit user-created simple databases during a logs only backup.
    The Microsoft SQL Server does not support logs only backups for simple restore model databases. During a logs only backup that includes simple restore model databases, the simple databases are backed up at level full. This option does not affect the backup of system databases, such as master and msdb, which are promoted to level full during a logs only backup and cannot be omitted from the backup.

    This option does not apply to full and cumulative incremental backups.

    • Select **Turn off backup promotion** to disable backup promotion.

    • Select **Skip databases that are in an incompatible state**—Select to skip databases that are not compatible for backup.

    • In the **DBCC Consistency Check Options** area, select the appropriate options:

      ▪ Select **Select all databases** to run database consistency checks (DBCC) on all of the databases. If the DBCC check fails, the database backup is skipped and backup status of the group appears as failed. This action ensures that all backed-up databases are suitable for restore.

      ▪ In the **Databases** box, select specific databases for the DBCC check.

    • In the **Striping Options** area, specify striping options:

      a. To enable striping during a backup, select **Enable Striping**.

      b. In the **Specify the number of stripes** field, type or select the number of stripes.

    The following figure shows the Client Configuration wizard **Specify the Backup Options** page.

Figure 16 Specify the Backup Options page

14. Click **Next**.

    The **Client Configuration Summary** page appears.

15. On the **Client Configuration Summary** page, check to ensure that the configuration choices are correct. Click **Back** to make changes, or click **Create** to configure the client resources.

    The **Check Results** page appears.

16. On the **Check Results** page, review the messages to ensure that the client was successfully created. You should see the following message:

    ```
    Successfully added new client "client_name"
    ```

17. Click **Finish** to exit the wizard.

After you finish

Open the NetWorker **Administration** window to the **Protection** tab. Click **Clients** in the expanded left pane and ensure that the newly created client is listed.

The following figure shows the NetWorker **Administration** window open to the **Protection** tab.

Figure 17 Viewing clients in the Protection tab
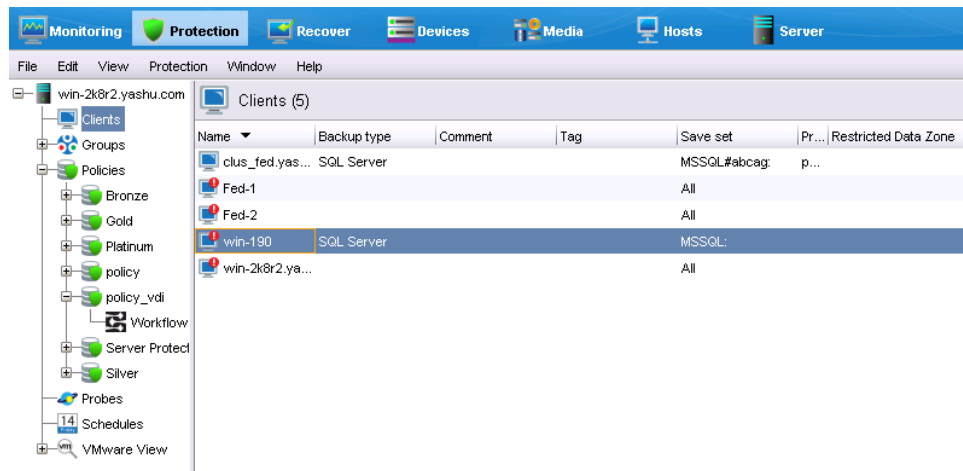
> (i) **Note:** To make updates to an existing client resource, right-click the client resource and select **Client Backup Configuration** > **Modify Client Wizard**.

Configuring a client resource by using the Client Properties dialog box

You can manually create a client resource by using the **Client Properties** dialog box of the **NetWorker Administration** window. The *NetWorker Administration Guide* and the *NetWorker Module for Microsoft Administration Guide* provide details about data protection policy. Review the information in these guides before manually creating the client resources.

About this task

When you manually create client resources for cluster configurations, ensure that a client resource is created for Windows cluster name and that dummy client resources are created for the other participating nodes in the cluster. If you do not create dummy clients for each of the nodes in the cluster, the backup fails. Do not assign a group for the dummy clients.

Click the question mark icon at the lower left of each page for details about each field in the page.

Procedure

1. In the **Administration** window, click **Protection**.

2. In the expanded left pane, select **Clients**.

3. From the **File** menu, select **New**.

4. On the **General** tab:

   - In the **Name** field, type the fully qualified domain name (FQDN) of the SQL Server, Windows cluster, or availability group listener. If you create multiple client resources for the same SQL Server, use the same name for each.
     > (i) **Note:** Do not type the IP address as the client name.

   - In the **Comment** field, type a description. If you create multiple client resources for the same NetWorker client host computer, use this attribute to differentiate each resource's purpose.

   - In the **Tag** field, type one or more tags to identify this client resource for dynamic client groups in data protection policies.

   - In the **Retention Policy** field, select a retention policy from the list.
     The retention policy determines the time period during which the rolled-over data is available.

- Leave the **Block based backup** option clear. This feature is not supported for SQL Server VDI.

- From the **Directive** list, select an option.
  Directives are resources that contains special instructions that control how the NetWorker server processes files and directories during backup.

- Select **Scheduled Backups**.
  The **Backup type** field displays the SQL Server.

- In the **Save Set** field, specify the components to be backed up as listed in the following tables.
  (i) Note: When nodes within an availability group (AG) reside on SQL server instances with different names, any instance name can be used. NMM will automatically detect the AG that resides on the instance name.

Table 22 Save sets in a standalone environment

Save set	Description
All	Default value.
MSSQL:	Backs up all the databases on the SQL Server host or the SQL default instance.
MSSQL$<SQL_named_instance>:	Backs up the specified SQL named instance.
MSSQL $<SQL_named_instance>:dbName	Backs up the database of the specified SQL named instance.
MSSQL:dbName1 [MSSQL:dbName2 MSSQL:dbName3 ...]	Backs up the specified databases from the SQL default instance.

Table 23 Save sets in a cluster environment

Save set	Description
All	This save set for the physical nodes is the default value.
MSSQL#<AlwaysOn_Availability_Group_name>:	This save set for the listener client backs up all the databases of the AlwaysOn Availability Group that is configured with the SQL default instance.
MSSQL $<SQL_named_instance>#<AlwaysOn_Availability_Group_name>:	This save set for the listener client backs up all the databases of the AlwaysOn Availability Group that is configured with the SQL named instance.

- Select the appropriate option in the **Protection group list** field.
  If client resources for the same NMM host are added to different backup groups, ensure that the **Start Time** field for each backup group is spaced such that the backups for the host's client resources do not overlap.

- In the **Schedule** field, select a backup schedule.

5. On the **Apps & Modules** tab:

- In the **Remote user** and **Password** fields respectively:

  - For SQL Server 2012 or later, type the remote username and password in the **Remote User** and **Password** fields. To enable NMM to back up the SQL Server virtual server or a mirrored server, type the username for a Windows user account that has SQL Server administrator privileges.

  - For other SQL Server versions, leave the fields empty.

- In the **Backup command** field, type the `nsrsqlsv` command and any necessary command options.
  Command options for nsrsqlsv on page 63 provides the supported backup command options.

- For **Application Information** field, type the required values. The following table describes the available application information variables.

Table 24 Application Information field values

Values	Default and valid values
ENABLE_GLR	• **TRUE** (Default): Enables granular-level recovery (GLR) capable backups. • **FALSE**: Prevents the backup from being GLR-capable. This setting is useful if you are using technology that is not supported for GLR (such as compression, encryption, or tape-type devices) and want to avoid error messages in the operation logs.
NSR_INCLUDE_AAG_DATABASE	• **AAG_ANYREPLICA** (Default): Includes Always On availability group databases in instance-level backup operations. • **None**: Omits Always On availability group databases from instance-level backup operations.
NSR_SKIP_SIMPLE_DB	• **TRUE**: Omits user-created simple databases during a logs only backup. This option does not affect the backup of system databases, such as master and msdb, which are promoted to level full during a logs only backup and cannot be omitted from the backup. • **FALSE** (Default): Includes simple databases in the backup. During a logs only backup that includes simple recovery model databases, the simple databases are backed up at level full. This option does not apply to full and cumulative incremental backups.
NSR_SKIP_NON_BACKUPABLE_STATE _DB	• **TRUE**: Omits databases that are in not in a state to be backed up from the backup operation. The savegroup reports success for the SQL instance level backups. • **FALSE** (Default): Includes databases that are in not in a state to be backed up from the backup operation. The backup of these databases fail and the savegroup reports failure of the SQL instance level backups.
NSR_BACKUP_PROMOTION	• **ALL**: Enables backup promotion to occur in any applicable scenario. • **NONE**: Disables backup promotion. Logs a warning when backup promotion would normally occur. • **NONE_WITH_WARNINGS**: Disables backup promotion. Logs a warning when backup promotion would normally occur. • **SKIP_RECOVERY_MODEL**: Disables database recovery model change detection. Backup promotion as a result of recovery model change will not occur, but backup promotion in other scenarios will still occur.
NSR_MIRROR_INSTANCE_PORT=*<port number>*	Specifies the port number when a SQL mirror database is configured with custom TCP port.

- Select **None** in **Proxy Backup**.

- For data deduplication using a Data Domain device, under **Deduplication**, select **Data Domain backup**.

6. On the **Globals (1 of 2)** tab, complete fields, as required.

7. For federated backups, type all known aliases for the SQL Server host as in the following example:

   ```
   mars mars.emc.com
   ```
   Include both the DNS short name and long name for the SQL Server host.

8. On the **Globals (2 of 2)** tab, in the **Remote Access** field, type the user IDs or hostnames of other clients.

   This field grants copy restore type permissions, which enables the named hosts to access the NetWorker server and receive directed recover data. If this field is empty, only administrators and users who are logged in to the SQL Server host have access.

   For mirroring, this should be the same user account and password that you used to set up the mirroring relationship. If multiple accounts have been set up, only one must be specified. According to Microsoft SQL documentation, a domain account must be used to set up the mirroring relationship.

9. Click **OK**.

10. Run the backup from the savegroup.

Configuring cluster client resources with the Client Properties dialog box

When you create client resources manually for cluster configurations, ensure that a client resource is created for Windows cluster name and that dummy client resources are created for the other participating nodes in the cluster. If you do not create dummy clients for each of the nodes in the cluster, the backup fails. Do not assign a group for the dummy clients.

Creating client resources for each cluster node
When you create client resources, edit the client resource for each physical node of the cluster. In the NetWorker Administration GUI, in the left pane, select **Clients**, and then in the right pane, right-click to create a new resource.

About this task

On the **General** tab, in the **Name** field, type the fully qualified domain name for the cluster node name. For example:

```
wash-2k.belred.emc.com
```

Creating client resources for a virtual server
You must create client resources for each virtual server in the cluster.

Procedure

1. Open the NetWorker Management Console.

   The **NMC Enterprise** window appears.

2. Right-click the server and select **Launch Application**.

   The NetWorker Administration GUI appears.

3. In the NetWorker Administration GUI, click **Protection**.

4. In the left pane, select **Clients**.

5. From the **File** menu, select **New**.

6. Click the **General** tab.

7. In the **Name** field of the client resource, type the short name for the SQL virtual server without the domain specification. This field should contain the name of the virtual server, not a node name.

8. On the **Apps & Modules** tab (for NetWorker), in the **Backup command** field, type the `nsrsqlsv` command with the necessary options.

   For virtual server backups, the `-A` *`SQL_virtual_server_name`* command option is required.

9. On the **Globals (2 of 2)** tab, in the **Remote Access** field, add entries similar to the following one to grant access to all physical nodes in the cluster.

   RemoteUser@physicalnode_hostname

   where:

   - *RemoteUser* is the account under which the backup will run.

   - *physicalnode_hostname* is the fully qualified domain name of the physical node.

   The **Remote Access** field enables the NMM server to access the cluster node to authenticate the computer (on which the virtual server is running) as an NMM client before any backup or restore operation begins. Follow this step for each virtual server client resource in the cluster.

10. On the **Apps & Modules** tab, in the **User Name** and **Password** fields, add the username and password, respectively, for a Windows user account that has both SQL Server administrator privileges and Windows administrator privileges. The **User Name** and **Password** fields enable NMM to back up the SQL Server virtual server. Follow these steps for each virtual server Client resource in the cluster.

11. Use NMC to start the backup group manually, or wait for the next scheduled backup to occur.

Configuring client resources for clusterless availability group listener and physical nodes of an AlwaysOn Availability Group

To perform AlwaysOn Availability Group backups by using a clusterless availability group listener, you must manually create client resources for the listener and the physical nodes of the AlwaysOn Availability Group by using the **Client Properties** dialog box of the **NetWorker Administration** window.

About this task

(i) Note: Before you configure a client resource for the clusterless availability group listener, ensure that you review the basic information and the requirements in the Clusterless availability group listeners on page 39 section.

Procedure

1. In the **Administration** window, click **Protection**.

2. In the expanded left panel, select **Clients**.

3. From the **File** menu, select **New**.

4. On the **General** tab:

   - In the **Name** field, type the short name of the clusterless availability group listener client.

   - In the **Comment** field, type a description.

   - In the **Tag** field, type one or more tags to identify this client resource for dynamic client groups in data protection policies.

   - In the **Retention Policy** field, select a retention policy from the list.

The retention policy determines the time period during which the rolled-over data is available.

- Leave the **Block based backup** option clear. NMM does not support block based backups of SQL Server VDI data.

- From the **Directive** list, select an option.
  Directives are resources that contains special instructions that control how the NetWorker server processes files and directories during a backup.

- Select **Scheduled Backups**.
  The **Backup type** field displays the SQL Server.

- In the **Save Set** field, type the AlwaysOn Availability Group save set as the following table lists:

Table 25 Save sets

Save set	Description
`All`	This save set for the physical nodes is the default value.
`MSSQL#<AlwaysOn_Availability_Group_name>:`	This save set for the listener client backs up all the databases of the AlwaysOn Availability Group that is configured with the SQL default instance.
`MSSQL $<SQL_named_instance>#<AlwaysOn_Availability_Group_name>:`	This save set for the listener client backs up all the databases of the AlwaysOn Availability Group that is configured with the SQL named instance.

- From the **Protection group list** field, select the appropriate option.
  If client resources for the same NMM host are added to different backup groups, ensure that the **Start Time** field for each backup group is spaced such that the backups for the host's client resources do not overlap.

- In the **Schedule** field, select a backup schedule.

5. On the **Apps & Modules** tab:

- In the **Remote user** and **Password** fields, type the remote username and password respectively.

- In the **Backup command** field, type `nsrsqlsv`.
  Command options for nsrsqlsv on page 63 provides the supported backup command options.

- In **Application Information** field, type the required application information variable. The following table lists the available application information variables.

Table 26 Application Information field values

Values	Default and valid values
`ENABLE_GLR`	- `TRUE` (Default): Enables granular-level recovery (GLR) capable backups. - `FALSE`: Prevents the backup from being GLR-capable. This setting is useful if you are using technology that is not supported for GLR (such as compression, encryption, or tape-type devices) and want to avoid error messages in the operation logs.

Table 26 Application Information field values (continued)

Values	Default and valid values
`NSR_INCLUDE_AAG_DATABASE`	• `AAG_ANYREPLICA` (Default): Includes Always On availability group databases in instance-level backup operations. • `None`: Omits Always On availability group databases from instance-level backup operations.
`NSR_SKIP_SIMPLE_DB`	• `TRUE`: Omits user-created simple databases during a logs only backup. This option does not affect the backup of system databases, such as master and msdb, which are promoted to level full during a logs only backup and cannot be omitted from the backup. • `FALSE` (Default): Includes simple databases in the backup. During a logs only backup that includes simple recovery model databases, the simple databases are backed up at level full. This option does not apply to full and cumulative incremental backups.
`NSR_SKIP_NON_BACKUPABLE_STATE_DB`	• `TRUE`: Omits databases that are in not in a state to be backed up from the backup operation. The savegroup reports success for the SQL instance level backups. • `FALSE` (Default): Includes databases that are in not in a state to be backed up from the backup operation. The backup of these databases fail and the save group reports failure of the SQL instance level backups.
`NSR_BACKUP_PROMOTION`	• `ALL`: Enables backup promotion to occur in any applicable scenario. • `NONE`: Disables backup promotion. Logs a warning when backup promotion would normally occur. • `NONE_WITH_WARNINGS`: Disables backup promotion. Logs a warning when backup promotion would normally occur. • `SKIP_RECOVERY_MODEL`: Disables database recovery model change detection. Backup promotion as a result of recovery model change will not occur, but backup promotion in other scenarios will still occur.
`NSR_MIRROR_INSTANCE_PORT=<port number>`	Specifies the port number when a SQL mirror database is configured with custom TCP port.
`NSR_PS_DEBUG_LEVEL=<number_1_through _9>`	Generates detailed logs that you can use to troubleshoot the backup issues. You can specify a value 1 through 9. The default value is **0** (zero).

- Select **None** in **Proxy Backup**.

- For data deduplication using a Data Domain device, under **Deduplication**, select **Data Domain backup**.

6. On the **Globals (1 of 2)** tab, in the **Aliases** field, type the DNS short name and the FQDN of the clusterless availability group listener each in a separate line.

ⓘ Note: Example entries of the DNS short name and the FQDN of the clusterless availability group listener:

```
mars

mars.emc.com
```

7. On the **Globals (2 of 2)** tab, in the **Remote Access** field, add entries similar to the following to grant access to all physical nodes in the AlwaysOn Availability Group.

    RemoteUser@physicalnode_hostname
    where:

    *   *RemoteUser* is the account, under which the backup is performed.

    *   *physicalnode_hostname* is the FQDN of the physical node.

    The **Remote Access** field enables the NMM server to access the physical node that is configured with the clusterless availability group listener in the AlwaysOn Availability Group, and authenticates the node as an NMM client before any backup or restore operation begins.

8. Specify the fields on the other tabs according to your requirement, and then click **OK**.

9. Configure client resources for the physical nodes in the AlwaysOn Availability Group.

    To configure a client resource for each physical node, repeat steps 1 through 8, but with the following changes:

    a. In step 4:

    *   In the **Name** field, type the short name of the physical node.

    *   In the **Save Set** field, type `All`.

    b. In step 6, in the **Aliases** field, type the DNS short name and the FQDN of the physical node each in a separate line.

10. Create a policy, a workflow, an action, and a group for the listener client that you have created.

    Setting data protection policies on page 93 provides information.

11. Start the workflow.

Setting data protection policies

About this task

(i) Note: NetWorker 8.2.x does not use data protection policies. If you are using NetWorker server 8.2.x and NMM 19.2, follow the steps provided in the *NetWorker Module for Microsoft for SQL VDI User Guide* version 8.2 SP1 to configure a NetWorker Group.

Complete the following tasks to set data protection policies for scheduled backups:

*   Create a protection policy
*   Create a workflow
*   Create an action
*   Create a protection group

Creating a policy

Policies provide an organizational container for the workflows, actions, and groups that support and define the backup, management, and system maintenance actions that you want to perform.

Procedure

1. Open the NetWorker Administration GUI. From the **Protection** tab, right-click **Policies** from the expanded left pane, and then click **New**.

   The **Create Policy** window appears.

2. In the **Name** field, type a name for the policy.

   The following figure shows the **Create Policy** window.

   Figure 18 Create Policy window

3. Click **OK**.

Results

The policy is created and the NetWorker Administration GUI opens to the page for the newly created policy.

After you finish

Add workflows, actions, and protection groups to the policy.

Creating a workflow

Workflows define the start time for a series of actions, the frequency in which the actions run, the order of actions in a sequence, and the protection group to which the workflow applies.

Before you begin

Create a policy.

Procedure

1. Open the NetWorker Administration GUI.

2. From the **Protection** tab, click the policy that you must create a workflow for.

The **Policy** page opens.

3. Click **Create a new Workflow**.

The **New Workflow** window opens.
The following figure shows the **New Workflow** window.

Figure 19 Creating a workflow for the policy

4. In the **Name** field, type a name for the workflow and either specify the workflow settings or accept the default settings.

5. Click **OK**.

Results

The workflow is created and the NetWorker Administration GUI opens to the page for the newly created workflow.

Creating an action within a workflow

Actions are the key resources in a workflow for a data protection policy and define a specific task, for example, a backup, clone, or snapshot.

Before you begin

Create a policy and a workflow within that policy.

Procedure

1. Open the NetWorker Administration GUI and click the **Protection** tab. Click the policy that you must create a workflow for.

2. Expand the policy in the left pane that contains the workflow you must create an action in, and click the workflow.

The **Workflow** page opens.

3. Click **Create a new Action**.

The Policy Action wizard opens.

4. On the **Specify the Action Information** page, in the **Name** field, type a name for the action.

5. In the **Comment** field, type a description for the action.

6. To ensure that the action runs when the policy or workflow that contains the action is started, select **Enabled**. To prevent the action from running when the policy or workflow that contains the action is started, clear this option.

7. From the **Action Type** list, select the action.

8. In the **Workflow** area, assign the action to a policy and workflow.

9. In the **Period** area, specify the backup schedule.

   (i) Note: NetWorker supports backup levels full, logs-only, and cumulative incremental with SQL Server VDI.

   The following figure shows the Policy Action wizard.

   Figure 20 Specifying action information in the Policy Action wizard

10. Click **Next**.

11. On the **Specify the Backup Options** page, specify backup properties for the action or accept the default properties, and click **Next**.

12. On the **Specify the Advanced Options** page, set the **Retries** field to 0, and then click **Next**.

13. On the **Action Configuration Summary** page, ensure that the correct selections are displayed.

    • If you must change any specifications, click **Back**.

    • If you are satisfied with the summary, click **Configure** to create the action.

14. Review the messages on the **Action Wizard Results** page and ensure that the action was successfully created.

15. Click **Finish** to exit the wizard.

Results

The action is created within the specified workflow and is displayed in the NetWorker Administration GUI, on the **Workflow** page.

Creating a protection group

Basic client groups define a static list of client resources for an action.

Before you begin

Create a policy, workflow, and action.

Procedure

1. Open the NetWorker Administration GUI. From the **Protection** pane, click the policy that you must create a protection group within.

   The **Policy** page opens.

2. Click the **Groups** tab. Right-click within the tab, and then select **New**.

   The following figure illustrates adding a group to a policy from the **Protection** pane.

   Figure 21 Adding a group to a policy from the Protection pane
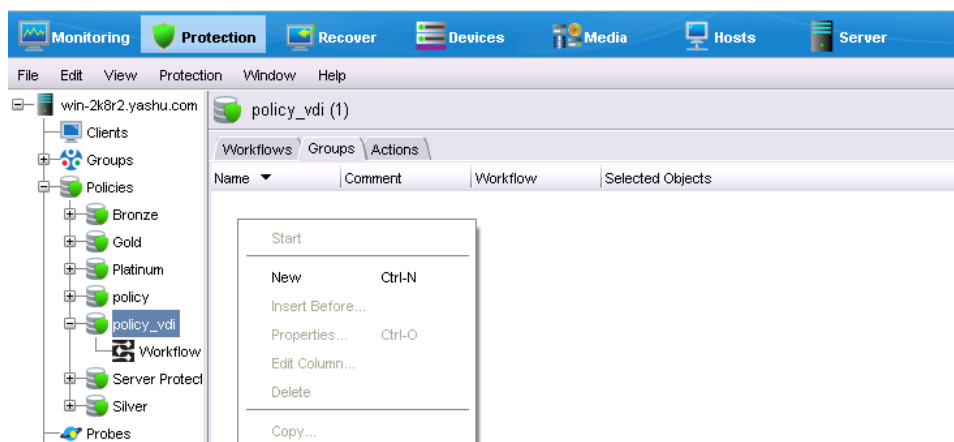
   The **Create Group** dialog box opens.

3. In the **Name** field, type a name for the group.

4. From the **Group Type** list, leave the default selection of **Clients**.

5. In the **Comment** field, type a description of the group.

6. Select the workflow in which to assign the group from the **Policy-Workflow** list.

   The following figure illustrates specifying a workflow and client in the **Create Group** window.

Figure 22 Specifying workflow and client in the Create Group window
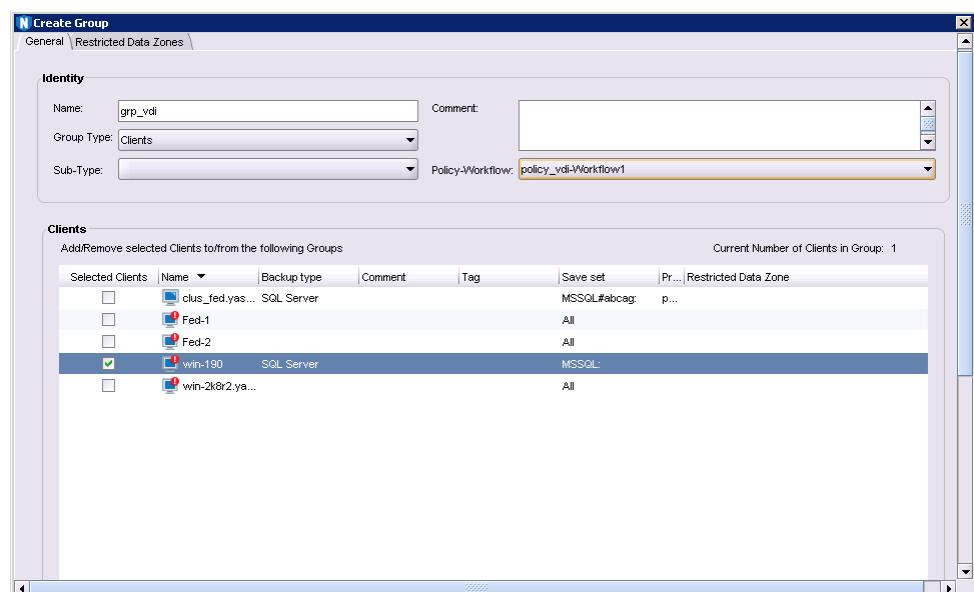
7. Click **OK**.

Results

The group is created and appears in the **Group** tab of the policy in the NetWorker Administration GUI.

Monitoring scheduled backups

During a backup, you can check for status messages in the NetWorker Administration GUI.

About this task

The NetWorker server reports the successful and failed database backups, including the databases in an Always On Availability Group. To monitor the success of a backup, use the following procedure:

Procedure

1. From the NMC, open the NetWorker Administration GUI.

2. Click the **Monitoring** tab.

3. Right-click a backup in the **Policies** pane, and then select **Show Details**.

   The **Show Messages** window appears.

Results

The **Show Messages** window displays successful and failed backup reports.
The following figure shows as example of a successful backup message.

Figure 23 Successful backup messages
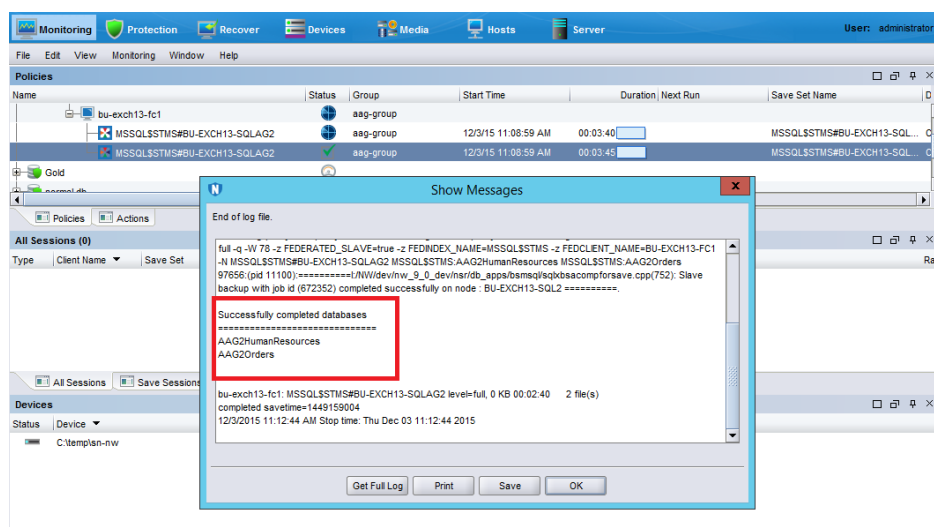
The following figure shows as example of a failed backup message.

Figure 24 Failed backup messages

Scheduled Backups
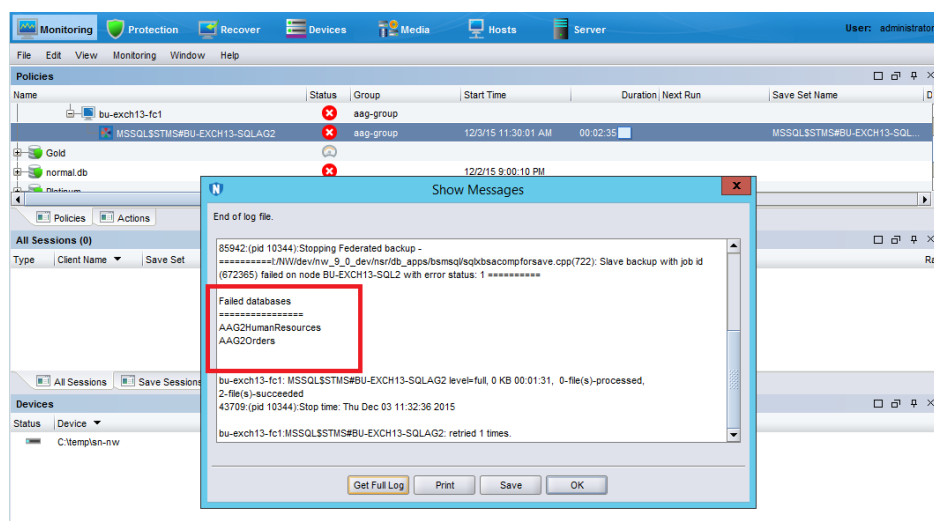
CHAPTER 6

Data Restore

This chapter includes the following sections:

Overview

You can restore SQL Server data by using one of the following methods:

- The NMM SSMS plug-in GUI
- The CLI — `nsrsqlrc` restore command

The Data Domain compressed restore feature is enabled by default.

A compressed restore uses less bandwidth by restoring the backed-up data in a compressed form from the Data Domain system to the SQL Server host. However, the feature is beneficial in a constrained bandwidth environment, but can impact the performance of the restore because of the usage of compression resources on the Data Domain system and the SQL Server host.

To disable the feature, perform the following steps:

1. On the SQL host, on which you perform the restore operation, go to the `\nsr\debug` folder.
2. Create the `disable_compressed_restore` file without any file extension.

(i) Note: Consider the following limitations:

- You cannot restore SQL Server data by using NMC.
- You cannot restore the data that was backed up by using the third-party tools.

Prerequisites

Review the prerequisites in this section before performing a recovery.

- Ensure that the user that is performing the recovery is assigned the required privileges, as defined in the section Access privileges for backup and recovery on page 44.
- Read the SQL Server product documentation to understand the limitations that are associated with recovery types on the various SQL Server versions.
- Ensure that the NetWorker server software is running on the host and the NetWorker Remote Exec Service is started on the SQL Server host.
- Ensure that no applications or services are waiting for or trying to log in to the SQL instance because restoring the SQL master database requires a restart of the SQL instance in single user mode followed by logging in to that instance. Single user mode allows only one administrator to log in.
- Wait for any database backups to finish. SQL Server will not restore a database while another database backup is in progress.
- Wait for any SQL Server startups to finish before starting a restore operation.
- Review the `ERRORLOG` file to determine if a database is being recovered, or search the `ERRORLOG` file for the **Recovery complete** string.
  If the `nsrsqlrc` program starts while the SQL Server is recovering databases, the following error message appears:

  ```
  Could not find database ID. Database may not be activated yet or may
  be in transition.
  ```

- Ensure that all database users are logged off the database. A restore fails if other users try to use the database during the restore operation.

> (i) **Note:** For the supported SQL Server versions, if the primary file group is not under restore, the online (piecemeal) restore functionality allows users to access a database while a backup or restore operation is in progress.

- Ensure that the SQL Server filestream feature is enabled on the recovery instance when you are recovering filestream data of SQL Server 2008 or later.

- Ensure that the primary file group is part of each piecemeal restore.
  SQL Server Enterprise Edition databases consisting of multiple file groups that can be restored in stages with piecemeal restore. The database remains offline during the restore process. Piecemeal restore is a multistage process that restores a database to itself or to another location. The initial stage must include the primary file group and optionally other file groups. After the primary file group is restored, you can bring the database online and continue restoring the remaining file groups in subsequent stages.

Restoring data by using the NMM SSMS plug-in GUI

Use the NMM SSMS plug-in GUI to perform SQL Server VDI restore operations.

About this task

In the **NetWorker** window, the **Script** view is available on each page. Click **Script** to generate the command prompt equivalent script. You can use the script to create a `.bat` file to perform scheduled backups, automation, and other tasks.

You can perform the following restores by using the **NetWorker** window in the NMM SSMS plug-in GUI:

- Copy restore

- Piecemeal restore

Procedure

1. In the SQL Server Management Studio, select the SQL Server instance that you want to restore, and then open the **NetWorker** window.

2. In the **NetWorker** window, on the **Database Restore** tab, under **Select a page**, click **General**.

   The **General** page appears.
   The following figure shows the **NetWorker** window open to the **Database Restore** tab, on the **General** page.

Figure 25 NetWorker dialog box—Database Restore tab General page
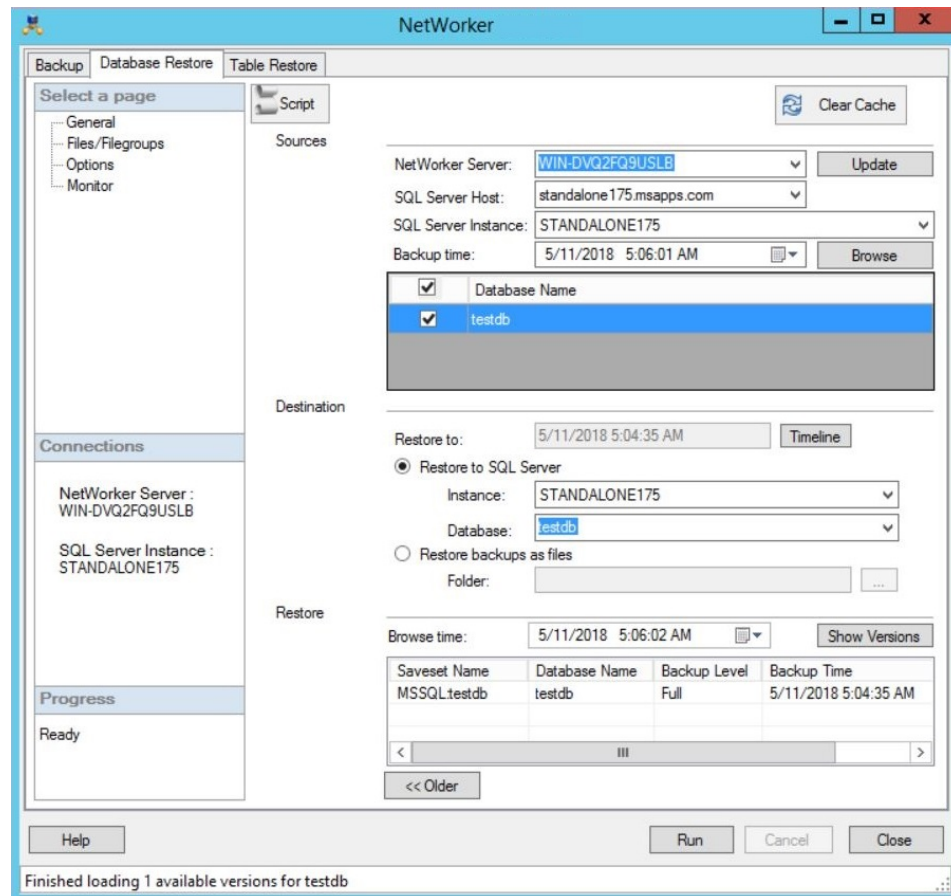
Loading icons appear while the information from the last restore is being loaded. While loading, you can specify other restore settings. Once the information is loaded, NMM populates all of the fields under **Sources**. To clear the saved settings, click the **Clear Cache** button.

3. Under **Sources**, make the following selections:

   a. From the **NetWorker Server** list, select the server that contains the backup.

   Click **Update** to update the list of available NetWorker servers.

   The **SQL Server Instance** list is refreshed after you select the NetWorker server.

   b. From the **SQL Server Host** list, select the client that contains the backup.

   If you used a cluster or an availability group listener to perform the backup, select the same cluster or listener name from the **SQL Server Host** list.

   c. From the **SQL Server Instance** list, select the server instance that contains the backup.

   Available backups are populated in the database list.

   d. From the **Backup time** list, select or browse for the time when the database backup was taken.

   e. From the database list, select one or more databases for restore.

4. Under **Destination**, choose the location where the backup is restored to:

   • To perform a restore directly to the database, select **Restore to SQL Server**, and then specify the destination:

a. From the **Instance** list, select the SQL Server instance.

(i) **Note:** If an availability group resides on multiple SQL Server instances, use the same SQL Server instance name that was specified when you configured the backup.

b. In the **Database** field:

- For an existing database, select the database that you want to restore from the list of backed-up databases.

- To perform a copy restore, specify the database name in the **Database** field. For example, in the following figure, `copy_c1_test_dbAG` is the copy restore name entered in that field. The backed-up database `c1_test_dbAG` will be restored by the name `copy_c1_test_dbAG`.

(i) **Note:** Redirected restore operations for multiple databases is not supported.

The following figure shows the **NetWorker** window open to the **Database Restore** tab, on the **General** page, with the database name for copy restore, `copy_c1_test_dbAG`, entered in the **Database** field.

Figure 26 NetWorker dialog box—Database Restore tab General page for copy restore
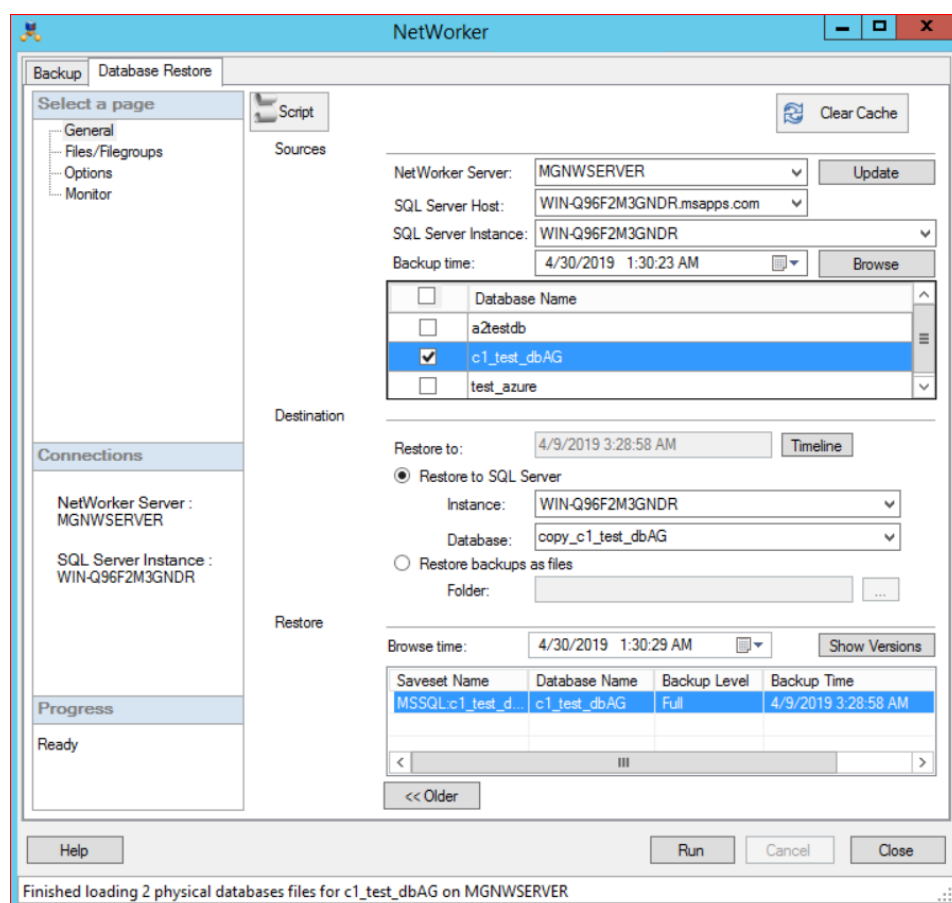
- To perform a flat-file restore, select **Restore backups as files**, and then in the **Folder** field, specify the destination for the files.

5. Under **Restore**, perform the following steps:

a. Specify the browse time through one of the following methods:

- From the **Browse time** list, select a time.

- Click **Show versions**, and then select a time.

  The table is populated with the save sets in the SQL database instance that are available for restore. The number of available backup versions appears in the bottom bar.

  b. From the table, select the save set that you want to restore.

6. To specify optional file settings, under **Select a page**, click **Files/Filegroups**.

   The **Files/Filegroups** page appears.

   The following figure shows the **NetWorker** window open to the **Database Restore** tab, on the **Files/Filegroups** page.

   Figure 27 NetWorker dialog box—Database Restore Files/Filegroups page
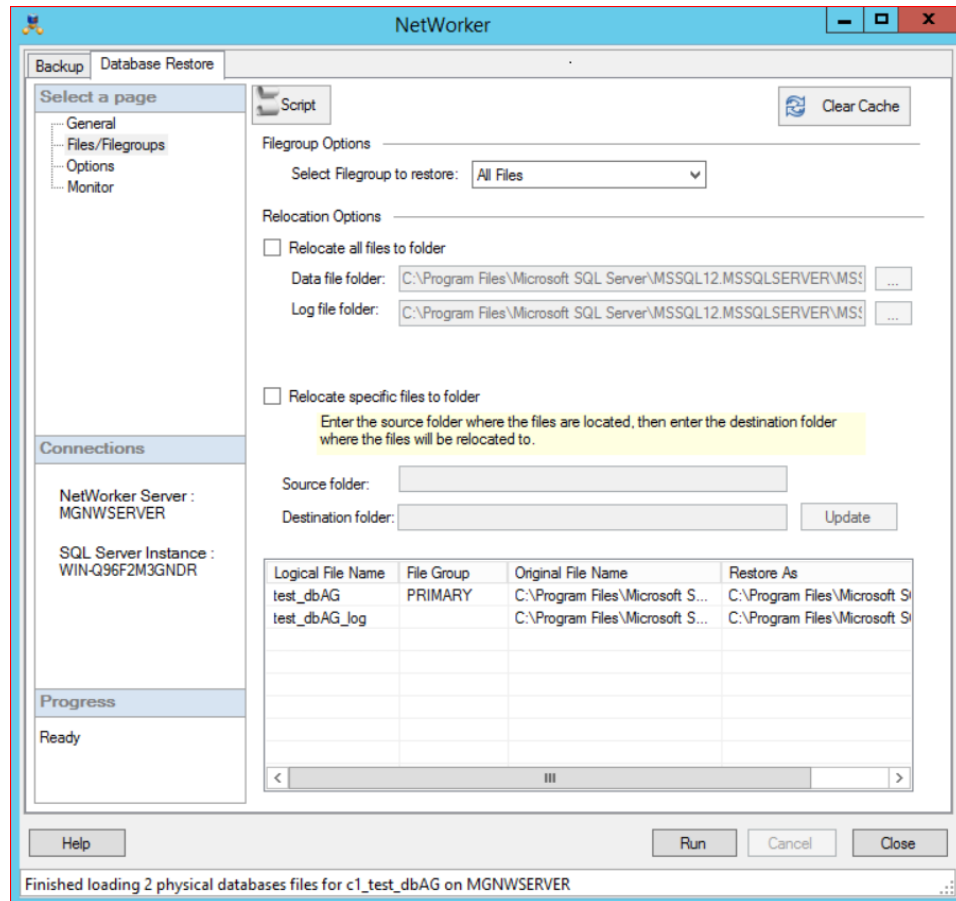
7. (Optional) To filter the list of available database files by file group, under **Filegroup options**, select a file group from the **Select Filegroup to restore** list.

   To list all available files, select **All files** from the list.

8. (Optional) To restore files to a different location, select a relocation option:

   - To relocate all datafiles to one folder and the log file to another folder, select **Relocate all files to folder**.

     Click the buttons on the right of **Data file folder** and **Log file folder** to specify the destination paths of the datafiles and the log file respectively.

     When you specify the target path for a copy restore, the operation keeps the original logical file name of the database.

   - To relocate specific files to a folder, where the operation can also change the logical names, select **Relocate specific files to folder**.

a. In the **Source folder** field, type the location path for the datafiles.

b. In the **Destination folder** field, type the destination path. The files are restored to this location.

c. Click **Update**.

Repeat these steps to relocate multiple file paths during a restore.

9. To specify additional restore options, under **Select a page**, click **Options**.

   The **Options** page appears.

   The following figure shows the **NetWorker** window open to the **Database Restore** tab, on the **Options** page.

   Figure 28 NetWorker dialog box—Database Restore Options page
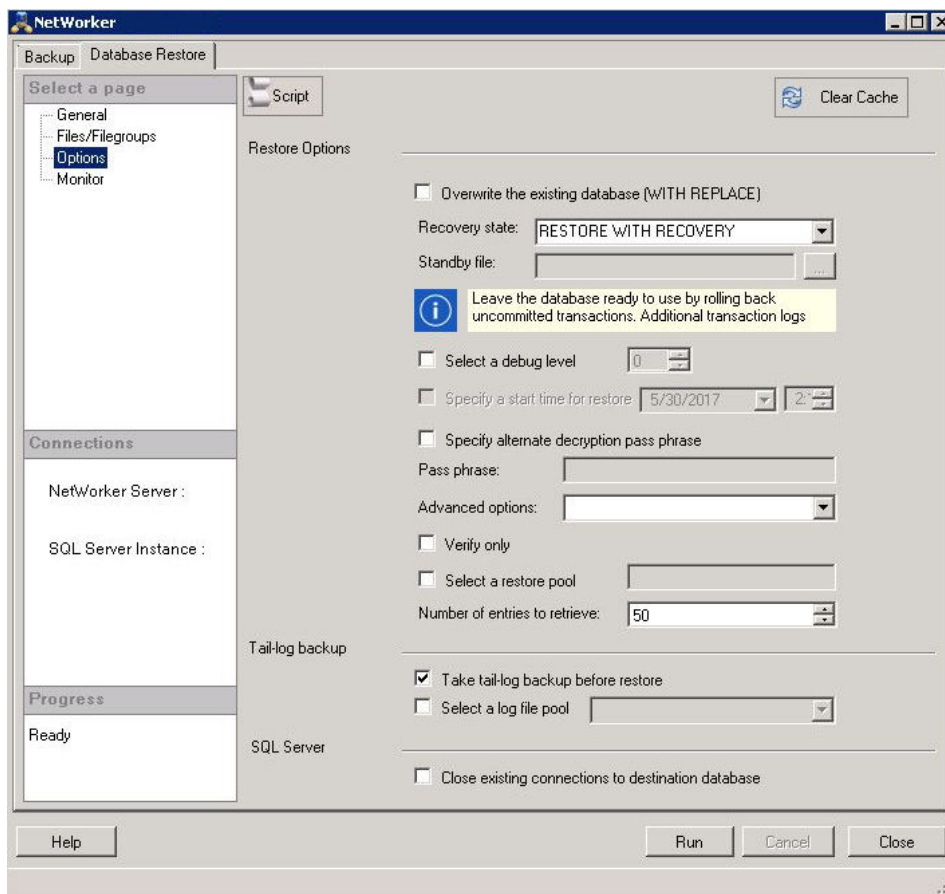
10. (Optional) Under **Restore Options**, select the following options as required:

   • To instruct the SQL Server to create the specified database and its related files, even if another database exists with the same name, select **Overwrite the existing database**. If another database exists with the same name, that database is deleted.

   • To specify a restore state, in the **Recovery state** field, select one of the following options:

      ▪ To instruct the SQL Server to leave the database in operational state after the restore, select **Normal mode**.
      Normal is the default mode.

      ▪ To activate the SQL Server NORECOVERY database restore option for the last stage restored, select **No Recovery mode**.

This mode places the database in a state that cannot be loaded after the restore. However, the database can still process additional transaction log restore operations.

- To enable the standby undo file feature for SQL Server to use when rolling back the transaction, select **Standby mode**.

- If the restore is in standby mode, from the **Standby File** field, browse and select an undo file.

- To select the level of information recorded in the logs that you can use for troubleshooting restore issues, select **Select a debug level**, and then select the troubleshooting level. Levels range 0-9, with 0 representing no information and 1 representing the least amount of information.

- To specify a timeframe for flat-file restore, select **Specify a start time for restore**, and then select a date and time.
  The date and time that is selected in this field is the start time for the restore period, and the end time is the time selected on the **General** page, from the **Browse time** list.

- To provide an alternate pass phrase, select **Specify alternate decryption pass phrase**. If the NMC pass phrase is unavailable, then the alternate pass phrase is used.

- To specify advanced restore options, in the **Advanced options** field, select or type advanced restore options. Separate multiple entries with a comma, for example:
  `BUFFERCOUNT=2, KEEP_CDC, KEEP_REPLICATION`

  The following table outlines the advanced restore options that NMM supports:

Table 27 Advanced restore options

Restore option	Description
`BUFFERCOUNT=number_of_IO_buffers`	This option specifies the total number of IO buffers that can be used during restore.
`KEEP_CDC`	This option enables change data capture (CDC) restore. When restoring a database with CDC enabled, the restore operation works differently depending on the restore destination. Use the to KEEP_CDC option to: ■ restore the CDC enabled database on the same SQL instance by overwriting an existing database. In this scenario, KEEP_CDC is optional. ■ restore the CDC enabled database with a different name on the same SQL instance. In this scenario, KEEP_CDC is required. ■ restore the CDC enabled database on a different SQL server instance. In this scenario, KEEP_CDC is required.
`KEEP_REPLICATION`	This option preserves the replication when restoring a published database. This option is required if a database was replicated when the backup was created.

- To verify that the backup is complete and readable, select **Verify only**.

- To specify the clone pool where the save set is restored, select **Select a restore pool**, and then type the name of the restore pool in the field.

- To change the number of entries that are retrieved from the server, in the **Number of entries to retrieve** field, select a number.
  The value that is specified in this field is the page size for number of backup versions that are listed for restore in the **Restore** table in the General view. This value also determines the number of entries that are retrieved when querying backup versions from the server.

The smaller the number that is specified, the faster the backup versions list is available. The minimum value is 1, the default value is 50, and the maximum value is 10000.

11. (Optional) Under **Tail-log backup**, select the following options as required:

   - To perform a logs-only backup of the data before you perform a restore operation, select **Take tail-log backup before restore**.
   A tail-log backup ensures that the operation backs up the data that has changed from the previous backup.

      (i) Note: NMM does not support tail-log backups when multiple databases are being restored. NMM also does not support tail-log backups for SQL AAG databases.

   - To specify a log file pool, select **Select the log file pool**, and then select the log file pool from the list of available pools.

12. (Optional) Under **Server connections**, select **Close existing connections to destination database** to ensure exclusive access to the database during the restore operation if multiple connections exist.

13. To start the restore operation, click **Run**.

   If you have not enabled the instant file initialization feature, a warning message appears before the restore operation starts. To continue with the restore, click **OK**. To abort the operation, click **Cancel**.

   See Microsoft documentation for details on enabling the database instant file initialization feature on the SQL Server.

Results

The restore operation runs.
To view the status of the restore operation, open the **Monitor** page.

The following figure shows the **NetWorker** window open to the **Database Restore** tab, on the **Monitor** page.

Figure 29 NetWorker dialog box—Database Restore Monitor page
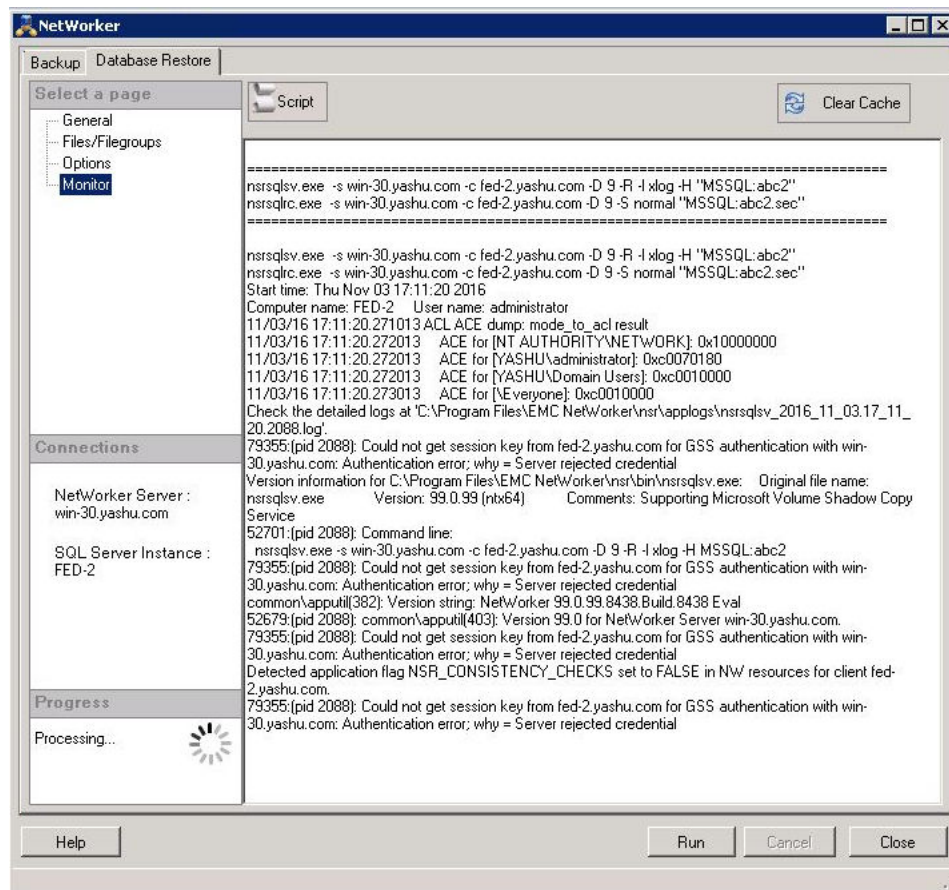
> (i) **Note:** If you do not have the instant file initialization feature enabled, the following warning message will appear in the output:

```
Instant file initialization is not enabled, please enable this feture for improved
restore performance.
```

However, the restore operation will continue.

Restoring data by using the CLI

Use the `nsrsqlrc` command to restore specified SQL Server data (files, file groups, and databases) from the NetWorker server.

To start a restore operation, specify the `nsrsqlrc` command and any of its options at a Windows command prompt.

Before you use the restore command, review the following considerations:

- Ensure that each command option is either lowercase or uppercase and, frequently both the cases of a letter are in the set of command options. Case is very important when specifying command-line flags.
  For example, `-c` specifies the NetWorker client name, while `-C` copies the database that is being restored to either the same SQL Server or a different SQL Server.

- Ensure that you use correct spacing. Depending on the command option, the space separator between an option and its corresponding argument can be optional, required, or not allowed. For example, both of the following expressions are valid:

```
-l backup_level
```

```
-lbackup_level
```

The following expression is invalid because a space is not allowed between the + argument and *log_mark_name*:

```
-M + log_mark_name
```

- Ensure that you use brackets ([ ]) to denote the optional portions of a command. When starting an actual backup or recovery operation, do not include brackets.

- Ensure that data items follow all other command options and parameters at the command prompt.

- If you do not have the instant file initialization feature enabled, the following warning message will appear in the output:
  ```
  Please ensure that instant file initialization is enabled on the
  destination SQL Server instance for improved restore performance.
  ```

  However, the restore operation will continue. See Microsoft documentation for details on enabling the database instant file initialization feature on the SQL Server.

Command syntax for nsrsqlrc

This section contains the command syntax for the `nsrsqlrc` command.

```
nsrsqlrc [-fjqVku]

[-$ instance_name]
[-a "option_key=value"]
[-A virtual_server_name]
[-b pool_name]
[-c client_name]
[-C file=path,file2=path2,...]
[-d MSSQL:destination_dbName]
[-e pass_phrase]
[-f]
[-H "'source_path';'destination_path'"]
[-j]
[-k]
[-M [+|-]log_mark_name]
[-O "Option1, Option2, Option3"]
[-P password]
[-q]
[-R fgName1,fgName2,...]
[-s NetWorker_server_name]
[-S normal|standby:undo_file|norecover]
[-t date]
[-u]
[-U username]
[-V MSSQL:dbName]
[-z]
[MSSQL: dbname dbname.fgName dbName.fgName.filename ...]
```

Command options for nsrsqlrc

The following table outlines the list of options available for the `nsrsqlrc` command.

Table 28 Command options for nsrsqlrc

Command options	Descriptions
`-$ MSSQL$ instance_name`	Specifies a named SQL Server instance as the source of the copy recovery.
	The following example copies the Sales database from the SQL Server prod-sql, instance Venus, to the SQL Server test-sql, instance Mars:
	`nsrsqlrc -s nw_server.company.com`
	`-c prod-sql. company.com`
	`-d MSSQL$MARS:`
	`-$ MSSQL$VENUS:`
	`-C " 'Sales '='D:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\Sales.mdf',`
	`'Sales_log'='D:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\Sales_log.ldf'"`
	`-t "Fri Dec 01 08:01:19 2006"`
	`"MSSQL$VENUS:Sales"`
	If the `-$` switch is used and no instance is named, for example "`-$ MSSQL:`", or the `-$` switch is omitted, the default instance is used.
`-a "RESTORE_TO_SOURCE_PATH=TRUE"`	Ensures that the backup is restored to the original path by default during a redirected restore operation. The `-a "RESTORE_TO_SOURCE_PATH=TRUE"` option is overridden if the `-C` or `-H` relocation options are used.
`-a "CLOSE_SQL_CONNECTIONS=TRUE"`	Ensures exclusive access to the target database by changing the database to single user mode during the restore operation to ensure a successful recovery.
`-a "FLAT_FILE_RECOVERY=TRUE"`	Performs flat-file recovery, which recovers the backup as files instead of restoring directly to the databases.
`-a "FLAT_FILE_RECOVERY_DIR=<destination_directory>"`	Specifies the directory where the recovery files are saved for flat-file recovery.
`-a "RESTORE_START_TIME=start_date"` `-t "end_date"`	Specifies a time range for flat-file recovery. Applicable backups within the time range are recovered. If the `RESTORE_START_TIME` variable is not defined when other flat-file recovery options are defined, required backups from the last level full backup until the `-t` time are recovered. If the last backup within the time range is a level differential backup, then all incremental backups in the time range are not

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions
	recovered. In this scenario, only full and differential backups are recovered to file. (i) **Note:** Follow the `nsr_getdate` command syntax guidelines when formatting the dates. See the *NetWorker Command Reference Guide* for details on the `nsr_getdate` command syntax.
`-A virtual_server_name`	Specifies the virtual server name when the SQL Server is configured to run as a clustered service in an MSCS cluster.
`-b pool_name`	Specifies the clone pool where the save set is restored from.
`-c client_name`	Specifies the NetWorker client name from which the SQL Server data is to be recovered.

`-C file=path,file2=path2,...`	Copies the database that is being restored to either the same SQL Server or a different SQL Server. This option can be used for normal, copy, and partial restores. Use the relocation list to specify new locations for recovered database files. The relocation list is composed of pairs of logical database file names and fully qualified domain database file name relocation paths. Specify the relocation list only when you restore a database. Each file name and relocation path is joined by the equal sign (=), and pairs are delimited in the list by commas.
	
	The syntax is:
	
	`["]['] file[']=[']path['],[']`
	`file2[']=[']path2['],...["]`
	
	For example, to copy the Project database from a client host that is named ClientHost1 to a different location on ClientHost1, specify the relocation list for the database files, but do not include the client hostname command option:
	
	`nsrsqlrc -s NetWorker_server_name`
	
	`-d MSSQL:CopyOfProject`
	
	`-C "'Project_Data'='C:\Relocation\Path`
	`\Project_Data.mdF',`
	
	`'Project_Data2'='C:\Relocation\Path`
	`\Project_Data2.ndF',...,`
	
	`'Project_Log'='C:\Relocation\Path`
	`\Project_Log.ldF MSSQL:Project'"`
	
	The relocation list must be enclosed by double quotation marks to allow for spaces in the relocation elements and path names. A logical file name or relocation path must be enclosed by single quotation marks to also allow for spaces. If a file name or path contains a single quotation mark, precede the single quote with another single quotation mark to prevent the NMM from parsing the single quotation mark as a delimiter, for example:
	
	`nsrsqlrc -s NetWorker_server_name`

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions
	`-d MSSQL:`*`CopyOfProject`* `-C `*`File'=C:\Relocate Dir\Path\,`* *`='C:\Relocate Dir\Path\',...,`* *`'=C:\Relocate Dir\Path\''" MSSQL:Project`* When no relocation list is specified, NMM reads the source database file names and location from the client index metadata or the backup. This information is used to generate a default list by relocating all files to the default SQL data path for the target SQL Server. The file names are guaranteed to be unique, but sufficient disk space is not ensured. To relocate an entire file path, see the table entry for the `-H` option. You cannot use the `-C` option with `-H`.
`-d MSSQL:`*`destination_dbname`* `MSSQL:`*`source_dbname`*	Performs a copy operation. The copy operation recovers SQL Server data from a client host to another database name on the same client host. The syntax is: `nsrsqlrc -s `*`NetWorker_server_name`*` -C `*`client_name`* `-d` `MSSQL:`*`destination_dbname`*` MSSQL:`*`source_dbname`* where: • *destination_dbname* is the name of the SQL database to which the source database is to be recovered. • *source_dbname* is the name of the SQL database to restore. When no relocation list is specified, NMM reads the source database file names and location from the client index metadata or the backup. This information is used to generate a default list by relocating all files to the default SQL data path for the target SQL Server. The file names are guaranteed to be unique, but it is not guaranteed that there will be sufficient disk space. When `-C`, `-M`, `-R`, or `-d` are used, the list of data items can include only one database. The list of data items must follow all other command options and parameters at the command prompt.
`-e `*`pass_phrase`*	Enables the use of an alternate pass phrase with AES encryption when recovering data.
`-f`	Performs a recovery operation by replacing the target database with the source. This option recovers a source database to an existing, incompatible database of the same name on the target host. This option is also used to recover damaged files.
`-H` `"'`*`source_path`*`';'`*`destination_path`*`'"`	Relocates file paths and all files contained in a path to a different location during a restore operation. The syntax for using this option is:

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions

	`nsrsqlrc -s `*`networker_server_name`*` -c `*`client_name`*` -H " '`*`source_path`*`';'`*`destination_path`*`'" -t "`*`MM/DD/YY HH:MM:SS`*`" -S `*`restore_mode`*` "MSSQL:`*`source_database`*`"`
	
	For example, if the testdb1 database has four datafiles that are in the `C:\datapath` source path and you want to relocate the database to the `D:\datapath` destination path, type the following:
	
	`nsrsqlrc -s networkerserver -c client -H "'C:\datapath';'D:\datapath'" -t "11/11/2015 10:30:57" -S normal "MSSQL $SQLSERVER2012:testdb1"`
	
	The `-H` command option can be used to relocate multiple file paths in the same command line. For example:
	
	`nsrsqlrc -s networkerserver -c client -H " 'C:\datapath1';'D:\datapath2','E:\datapath3';' F:\datapath4','G:\datapath5';'H:\datapath6' " -t "11/11/2015 10:30:57" -S normal "MSSQL $SQLSERVER2012:testdb1"`
	
	The `-H` option is supported on standard and redirected restores, which includes the following:
	
	• Normal restore—Same server and same instance.
	
	• Different instance restore—Same server and different instance.
	
	• Restore to different database file.
	
	• Different server restore.
	
	To relocate individual files, see the table entry for the `-C` option. You cannot use the `-C` option with `-H`.

`-j`	Performs a database consistency check between the backed-up SQL Server data and the recovered SQL Server data.
`-k`	Perform checksum before reading from media.
`-m `*`name`*	Specifies the cluster or availability group listener name.
`-M `*`[+|-]log_mark_name`*	Performs a SQL Server data recovery of the named transaction that is specified in `log_mark_name` (for SQL Server 2005 only). The way the mark name is prefixed determines how the data is recovered: • When the mark name is prefixed with a plus sign (+), the data is recovered to and includes the named transaction. • When the mark name is prefixed with a minus sign (–), the data is recovered up to the named transaction.

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions
	The `log_mark_name` should immediately follow the plus (+) or minus (-) symbol. The use of a space separator is not allowed. The default is the plus sign. For example, to recover the SQL data and include the named transaction `transaction_mark`, type the following command: **nsrsqlrc -s*NetWorker_server_name* -M +*transaction_mark* MSSQL:*dbName*** To recover the SQL data only to the named transaction `transaction_mark`, type the following command: **nsrsqlrc -s *NetWorker_server_name* -M -*transaction_mark* MSSQL:*dbName*** Specify only one SQL Server database, followed by all the other command options and parameters at the command prompt.
-O *"Option1, Option2, Option3"*	This option specifies advanced recovery options. The syntax is as follows: **-O "Option1, Option2, Option3"** You can specify the following advanced recovery options: • **BUFFERCOUNT=***number_of_IO_buffers* This option specifies the total number of IO buffers that can be used during a recovery operation. • **KEEP_CDC** This option enables change data capture (CDC) recovery. • **KEEP_REPLICATION** This option preserves the replication when you recover a published database. This option is required if a database was replicated when the backup was created. • **READ_WRITE_FILEGROUPS** This option recovers only the read/write (active) filegroups from the backup. This option can only be used to recover backups taken with the **READ_WRITE_FILEGROUPS** option. You cannot specify filegroup or file level recover targets or use the **-R** recover option with the **READ_WRITE_FILEGROUPS** option. After you restore a backup with the **READ_WRITE_FILEGROUPS** option, any read-only filegroups in the database will enter the recovery pending state and the rest of the filegroups will go online. There are two different scenarios to recover cumulative incremental **READ_WRITE_FILEGROUPS** backups: ■ **READ_WRITE_FILEGROUPS** This option recovers only the read/write (active) filegroups from the backup. This option can only be used

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions

	to recover backups taken with the **READ_WRITE_FILEGROUPS** option. You cannot specify filegroup or file level recover targets with the **READ_WRITE_FILEGROUPS** option.
	
	If you are relocating multiple filegroups during the restore, you can use the -H option for global relocation.
	
	After you restore a backup with the **READ_WRITE_FILEGROUPS** option, any read-only filegroups in the database will enter the recovery pending state and the rest of the filegroups will go online.
	
	There are two different scenarios to recover cumulative incremental **READ_WRITE_FILEGROUPS** backups:
	
	  — If you have taken a full **READ_WRITE_FILEGROUPS** backup prior to the cumulative incremental backup, the Microsoft application agent will chain together the two backups in the correct order and complete the restore.
	
	  — If you have not taken a full **READ_WRITE_FILEGROUPS** backup but want to use a normal full backup as the differential base for the cumulative incremental **READ_WRITE_FILEGROUPS** backup, you must first restore the full backup normally and leave the database in no recovery mode, and then apply the **READ_WRITE_FILEGROUPS** differential backup with the -z option.
	
	If you are specifying multiple options, separate each argument with a comma. For example:
	
	`nsrsqlrc -sNetWorker_server_name -O "BUFFERCOUNT=2, KEEP_CDC, KEEP_REPLICATION, READ_WRITE_FILEGROUPS" MSSQL:`

`-P password`	Specifies the SQL Server user password. When the `-U` username command option is specified, the password command option must also be provided, as follows: `nsrsqlrc -s NetWorker_server_name -U username -P password MSSQL:` Use the SQL Server username and password to log in to the SQL Server with SQL server authentication.
`-q`	Displays `nsrsqlrc` messages in quiet mode, which provides minimal information about the progress of the recovery operation, including error messages.
`-R fgName1,fgName2,...`	Performs a partial database recovery or a piecemeal recovery of the specified file groups. The partial database recovery operation restores specific file groups from a single full SQL Server database backup. Add the file groups to the `-R` command option

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions

	in a list, with items separated by commas. Use the `-C` parameter, and specify all files for the database. The `-d` option is also required:
	
	`nsrsqlrc -s NetWorker_server_name`
	
	`-R ["]['] fgName['],['] fgName2['],[']...['] ["]`
	
	`-C Project_Data=C:\Relocation\Path`
	`\Project_Data.mdF,`
	
	`Project_Data2=C:\Relocation\Path`
	`\Project_Data2.ndF,...,`
	`Project_Log=C:\Relocation\Path\Project_Log.ldF`
	`MSSQL:Project-d MSSQL:PartOfProject`
	`MSSQL:Project`
	
	where:
	
	• *fgName,...* are the names of the file groups to restore. Consider the following points when you list file groups to restore:
	
	■ To allow spaces between file group names, enclose the list of file group names within double quotes.
	
	■ Within file group names, enclose each file group name within single quotes.
	
	■ If a file group name contains a single quote, precede the single quote with another single quote to prevent the NetWorker client from parsing the single quote as a delimiter.
	
	For example, to accommodate for the space in file group A, the space after the first comma, and the single quote in file group A', use the following syntax:
	
	`-R "'Filegroup A', 'Filegroup A'''"`
	
	When an empty relocation list is supplied, use the following syntax:
	
	`-R ""`
	
	• *Project* is the name of the SQL database to restore.
	
	• *PartOfProject* is the name of the SQL database to which the source database is to be recovered.
	
	NMM recovers only the primary file group.
	
	When `-C`, `-M`, `-R`, or `-d` are used, the list of data objects can include only one database. The list of data objects must follow all other command options and parameters at the command prompt.

`-s NetWorker_server_name`	Specifies the NetWorker server to use for the recovery operation.

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions
`-S normal	standby:undo_file |` `norecover`	Performs the recovery operation according to the specified database restore mode of normal, standby, or no recovery. The syntax is: `nsrsqlrc -s NetWorker_server_name -` `ddestination_dbName -S` `normal	"standby:undo_file"	norecover` `MSSQL:source_dbName` where: • *destination_dbName* is the name of the SQL database to which the source database is to be restored. • *source_dbName* is the name of the SQL database to restore. The recovery modes are as follows: • The normal recovery mode recovers the database in normal, operational mode. • The standby recovery mode activates the SQL Server **STANDBY** option, which forces the database to be in a read-only state between transaction log recovery operations. • The norecover recovery mode activates the SQL Server **NORECOVER** option, which places the database in an unloadable state after the recovery. The database can still process additional transaction log recovery operations. For example, to recover the Project database in normal, operational mode to a new database named NewProjectName, type the following command: `nsrsqlrc -s NetWorker_server_name -S normal` `-d MSSQL: NewProjectName MSSQL:Project` To recover the database in standby mode, the standby parameter must be immediately followed by a colon, and the standby undo file location and file name must be specified. If a file name or location contains a space, enclose the file name or location within double quotes, for example: `nsrsqlrc -s NetWorker_server_name` `-S"standby:C:\temp\undo filename"` `-d MSSQL:NewProjectName MSSQL:Project`
`-t "MM/DD/YY HH:MM:SS"`	Recovers the SQL Server data on a specified date. If the date of a backup version is before or is equivalent to the specified date, the backup version is recovered. Follow the `nsr_getdate` command syntax guidelines when you format the date. See the *NetWorker Command Reference Guide* for details on the `nsr_getdate` command syntax.

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions
	To avoid adversely affecting the database, do not use the `-t` option if you are restoring a file or file group.
`-u`	Continues the recovery even if a checksum error occurs.
`-U` *username*	Specifies the SQL Server username. When the username command option is specified, the `-P` password command option must also be provided, for example: `nsrsqlrc -s NetWorker_server_name -U username -P password MSSQL:` Use the SQL Server username and password to log in to SQL Server with SQL server authentication.
`-V MSSQL:`*dbName*	Verifies the SQL Server database that is selected for the recovery. The `-V` command option verifies only that the selected database backup is suitable for restoring, the backup is not recovered. The syntax is: `nsrsqlrc -s NetWorker_server_name -V MSSQL:dbName`

`-z`	Enables implementation of a recovery plan to restore individual backups from a timestamp.
	By default, NMM builds the recovery plan, ensuring that all the required backups are available and that they run in the proper order and with the proper options. The `-z` option removes NMM safety checks.
	
	The `-z` option disables building the recovery plan and recovers only the backups that correspond to the specified timestamp. The `-z` command is required for all except the first (full) recovery command.
	
	Use the `-S norecover` option with the `-z` option to prepare the database for more restores. This command is required for all except the last recovery command.
	
	The final command recovers the last logs-only backup and brings the database online.
	
	The `-z` option is used in more complex recoveries. The example commands in this section use the following backup history:
	
	`savetime 1:00 — full`
	
	`savetime 2:00 — txnlog`
	
	`savetime 3:00 — txnlog`
	
	Consider the following example uses of the `-z` option:
	
	• To restore a database with a single command, specify the most recent backup timestamp or `now`. The following list provides examples:

Table 28 Command options for nsrsqlrc (continued)

Command options	Descriptions

	■ The following example uses the most recent timestamp:
	`nsrsqlrc ... -t "savetime 3:00" ...`
	
	■ The following example uses `now`:
	`nsrsqlrc ... -t "now" ...`
	
	These commands restore the entire backup chain from the last backup, regardless of the backup level.
	
	• To restore a database in a series of independent commands, type a separate command for each backup. You must recover the backup chain in order. For the final command, specify the most recent backup timestamp or `now`. The following list provides examples:
	
	■ The following example uses the most recent timestamp for the final command:
	`nsrsqlrc ... -z -S norecover -t "savetime 1:00" ...`
	
	`nsrsqlrc ... -z -S norecover -t "savetime 2:00" ...`
	
	`nsrsqlrc ... -z -S normal -t "savetime 3:00" ...`
	
	ⓘ Note: You can specify a point-in-time restore within the last logs-only backup by replacing the save time with the point-in-time. For instance, in the previous example, you can replace 3:00 with 2:45.
	
	■ The following example uses `now` for the final command:
	`nsrsqlrc ... -z -S norecover -t "savetime 1:00" ...`
	
	`nsrsqlrc ... -z -S norecover -t "savetime 2:00" ...`
	
	`nsrsqlrc ... -z -S normal -t "now" ...`
	
	The database is not available for general use until after the final recovery completes. Any missing, incorrect, or out-of-order save times result in SQL Server reporting errors.

Backup and recovery command syntax for SQL Server data

Use the additional command syntax that is shown in the following table to back up or restore SQL Server data with the `nsrsqlsv` and `nsrsqlrc` backup and recovery commands.

You can specify more than one data object and can combine different types of data. You must specify the SQL data objects with the syntax shown in the following table.

Table 29 Command syntax for SQL Server data

SQL Server data	Syntax for SQL Server data objects
All databases in the SQL Server storage hierarchy (optional)	`MSSQL:` Typing MSSQL: yields an instance-level backup of all databases on the SQL Server host.

Specified databases	`MSSQL:dbName`
	
	or
	
	`[MSSQL:dbName MSSQL:dbName2 ...]`

All file groups in specified databases	`MSSQL:dbName`
	
	or
	
	`[MSSQL:dbName. MSSQL:dbName2 ...]`

Specified file groups in specified database	`MSSQL:dbName.fgName`
	
	or
	
	`[MSSQL:dbName.fgName MSSQL:dbName.fgName2`
	`MSSQL:dbName2.fgName MSSQL:dbName2.fgName2 ...]`

Specified files in specified file groups in specified databases	`MSSQL:dbName.fgName.filename`
	
	or
	
	`[MSSQL:dbName.fgName.filename`
	`MSSQL:dbName.fgName2.filename`
	`MSSQL:dbName2.fgName.filename`
	`MSSQL:dbName2.fgName2.filename ...]`

Specifying `MSSQL` before each data object name is optional and does not affect the expression or the resulting operation. However, when `MSSQL` is specified, it must be followed by a colon (:).

For example, the following two commands are equivalent:

```
nsrsqlsv -s NetWorker_server_name dbName.fgName
nsrsqlsv -s NetWorker_server_name MSSQL:dbName.fgName
```

In a non-clustered, named instance configuration, `MSSQL$` is required, followed by the instance name and a colon. For example:

```
nsrsqlsv -s NetWorker_server_name MSSQL$instanceName:dbName.fgName
```

Syntax for a named instance configuration

When a configuration contains non-clustered named instances of the SQL Server, you must specify the name of the instance before the data.

```
MSSQL$instanceName:[dbName ...] [.fgName ...] [.fileName ...]
```

(i) Note: The `nsrsqlsv` and `nsrsqlrc` commands support specification of only a single instance. If save sets for more than one instance are specified, the backup fails. The `nsrsqlrc` command supports mixing instances for a copy restore operation.

Example 6 Back up all databases in an instance

To back up all the databases for instanceOne, type the following:

```
nsrsqlsv -s NetWorker_server_name MSSQL$instanceOne:
```

Example 7 Recover several file groups in an instance

To recover several file groups for instanceTwo, type the following:

```
nsrsqlrc -s NetWorker_server_name MSSQL$instanceTwo:dbName.fgName
MSSQL$instanceTwo:dbName.fgName2
```

Example 8 Using clustered instance SQL Server virtual server names with -A or -c

Instead of using clustered named instances in this syntax, use clustered instance SQL Server virtual server names with -A or -c options, as shown in the following command:

```
nsrsqlsv -s NetWorker_server_name -A SQL_virtual_server_DNS_name -c
SQL_virtual_server_DNS_name MSSQL:
```

where:

- *NetWorker_server_name* is the hostname of the NetWorker server.
- *SQL_virtual_server_DNS_name* is the Domain Name System (DNS) name for the SQL Server virtual server.

Create a client resource under the SQL Server virtual server DNS name.

For scheduled backups of a SQL Server virtual server client, you do not need to specify the −A or −c option with the SQL Server virtual server name. The savegrp process automatically specifies the virtual server name to the nsrsqlsv process by using the -m option.

Syntax for names containing a period, backslash, or colon

NMM provides command line syntax that enables you to back up and recover file names, file groups, and databases containing a period (.), backslash (\), or colon (:). By typing a backslash before the period or backslash, the nsrsqlsv and nsrsqlrc commands interpret the period or backslash as a literal character.

The tables in this section list the syntax for file names, file groups, and databases containing a period, backslash, colon, or any combination of the three.

The following notes apply to the information in the tables:

- The syntax that is shown in the right column applies to both the nsrsqlsv and nsrsqlrc commands.
- The notation MSSQL: is optional only for the nsrsqlsv command.
- A single period (.) continues to delimit SQL identifiers.
- The syntax also applies to named instances.
- The backslash period (\.) character sequence replaces each literal period in the SQL identifier.
- The double backslash (\\) character sequence replaces each literal backslash in the SQL identifier.

The following table lists the syntax for file names, file groups, and databases containing a period.

Table 30 Command syntax for names containing a period

Name visible from SQL utilities	Equivalent command-line syntax
SQL database named MyDatabase.COM	`MSSQL:MyDatabase\.COM` `MSSQL$MyInstance:MyDatabase\.COM`

SQL file group named MyFileGroup.2 for the SQL database named MyDatabase.COM	`MyDatabase\.COM.MyFileGroup\.2`
	`MSSQL:MyDatabase\.COM.MyFileGroup\.2`
	
	`MSSQL$MyInstance:MyDatabase`
	`\.COM.MyFileGroup\.2`

SQL file named MyFile.2, which is a member of the SQL file group named MyFileGroup.2 for the SQL database named MyDatabase.COM	`MyDatabase\.COM.MyFileGroup\.2.MyFile\.2`
	`MSSQL:MyDatabase\.COM.MyFileGroup`
	`\.2.MyFile\.2`
	
	`MSSQL$MyInstance:MyDatabase\`
	`\COM.MyFileGroup\\2.MyFile\\2`

The following table lists the syntax for file names, file groups, and databases containing a backslash.

Table 31 Command syntax for names containing a backslash

Name visible from SQL utilities	Equivalent command-line syntax

The SQL database named MyDatabase\COM	`MyDatabase\\COM`
	`MSSQL:MyDatabase\\COM`
	
	`MSSQL$MyInstance:MyDatabase\\COM`

The SQL file group named MyFileGroup\2 for the SQL database named MyDatabase\COM	`MyDatabase\\COM.MyFileGroup\\2`
	`MSSQL$MyInstance:MyDatabase\`
	`\COM.MyFileGroup\\2`

The following table lists the syntax for file names, file groups, and databases containing a colon.

Table 32 Command syntax for names containing a colon

Name visible from SQL utilities	Equivalent command-line syntax

SQL database named MyDatabase:COM	`MyDatabase:COM`
	`MSSQL:MyDatabase:COM`
	
	`MSSQL$MyInstance:MyDatabase:COM`

SQL file group named MyFileGroup:2 for the SQL database named MyDatabase:COM	`MyDatabase:COM.MyFileGroup:2`
	`MSSQL:MyDatabase:COM.MyFileGroup:2`
	
	`MSSQL`
	`$MyInstance:MyDatabase:COM.MyFileGroup:2`

SQL file named MyFile, which is a member of the SQL file group named MyFileGroup:2 for the SQL database named MyDatabase:COM	`MSSQL`
	`$MyInstance:Mydatabase:com.MyFileGroup:2.`
	`MyFile`

Example recovery command syntax

Example syntax to restore federated backups of a SQL Always On Availability Group

In the following example commands, the source backup object name, SQ12INST4 is the SQL server instance name used during the backup.

- The following command restores the federated backup to the same database:
  ```
  nsrsqlrc.exe -s nwsrvr -c sql2012clus3.brsvlab.local -f -S normal -$ "MSSQL
  $SQ12INST4:" -d "MSSQL$SQ12INST4:database1" "MSSQL $SQ12INST4:database1"
  ```

- The following command restores the federated backup to a different instance and database (redirected restore)
  ```
  nsrsqlrc.exe -s nwsrvr -c sql2012clus3.brsvlab.local -C"
  'testdb1_Data'='E:\sql2012_data\database1.mdf', 'testdb1_Log'='F:\sql2012_log
  \database1_log.LDF'" -f -S normal -$ "MSSQL$SQ12INST4:" -d "MSSQL
  $SQ12INST5:copy-database1" "MSSQL$SQ12INST4:database1"
  ```

- The following command restores multiple databases within an Always On Availability Group:
  ```
  nsrsqlrc.exe -s nwsrvr -c sql2012clus3.brsvlab.local -f -S normal -$ "MSSQL
  $SQ12INST4:database1" "MSSQL$SQ12INST4:database2"
  ```

Example syntax to perform a piecemeal recovery in an active mirror session

The following lines provide sample commands for a piecemeal recovery of the primary file group (mdf & ldf) and file group "a" (ndf) of AcmeBank to the new database AcmeOnline:

```
nsrsqlrc -s "bv-v-cgd2.belred.legato.com" -c "bv-v-cgd2.belred.legato.com" -$
"MSSQL$THREE:" -R "'PRIMARY','a'" -d "MSSQL$THREE:AcmeOnline" -C
"'AcmeBank'='E:\Data\AcmeOnline.mdf', 'AcmeBank_log'='E:\Data
\AcmeOnline_log.ldf', 'AcmeBank1'='E:\Data\AcmeOnline1.ndf'" -t "Wed Sep 14
13:31:46 2005" "MSSQL$THREE:AcmeBank"
```

(i) **Note:** The AcmeOnline database name and file locations are different from AcmeBank.

Data Restore

CHAPTER 7

Granular Level Recovery

This chapter includes the following sections:

Overview

NMM supports GLR of SQL Server backups that are created by using NMM and SQL VDI.

GLR enables you to restore specific items from a single backup without having to restore the entire backup. By using GLR with SQL Server, you can restore individual tables to the production database. This feature reduces the space requirements on local system storage during a restore operation. Depending on the size of the content database, GLR may also reduce restore time.

NMM SQL GLR requires GLR enabled backups. NMM SQL GLR enabled backups are enabled by default for NMM 9.1 and later. GLR enabled backups perform additional inline processing during a backup operation, and store additional metadata on the save set backup records. GLR enabled backups are not supported for direct backups to tape devices.

NMM SQL GLR is managed from the Table Restore tab in the NMM SSMS plug-in GUI. The Table Restore tab allows you to select the database and backup version for GLR. The Table Restore tab constructs a list of save sets that are contained in the selected backup version. NMM uses a mount service and the NetWorker Virtual File System (NWFS) to mount all save sets in the selected SQL backup version. The NWFS exposes the save sets as SQL native backup files on a virtual file system on the NMM client.

After the Table Restore tab mounts all save sets as a virtual file system, ItemPoint™ for Microsoft SQL Server launches. ItemPoint reads the NWFS and presents the database and tables that are contained in the backup, enabling you to restore SQL tables to the production database. Once GLR is performed through ItemPoint and the program is exited, the NMM SSMS plug-in GUI unmounts the backup.

The NMM SSMS plug-in GUI mounts SQL backups as NWFS by using a mount service. The mount service enables lifecycle management of the mount. While a backup is mounted, the mount service system tray icon appears. You can use the tray icon to inspect the mount path, extend the mount time, and unmount the backup.

Considerations

ItemPoint environmental requirements

Before using ItemPoint for Microsoft SQL Server, consider the environmental requirements listed in the following table:

Table 33 ItemPoint for SQL Server requirements

Component	Requirement
Operating systems	The following operating systems are supported: • Windows Server 2008 • Windows Server 2008 R2 • Windows Server 2012 • Windows Server 2012 R2 • Windows Server 2016 • Windows 7 • Windows 8 • Windows 10

Table 33 ItemPoint for SQL Server requirements (continued)

Component	Requirement
	ⓘ Note: 64-bit versions, virtual and physical, of the listed operating systems are required. Dell EMC ItemPoint for Microsoft SQL Server can only be run by users with administrative privileges and in administrative mode.
Microsoft SQL Server	Supported source file formats: • Native SQL backup 　■ Full, Differential, and Transaction Log backups 　■ Compressed and uncompressed • Offline SQL database files (.MDF/.NDF/.LDF) Supported source data: • SQL 2008 • SQL 2008 R2 • SQL 2012 • SQL 2014 • SQL 2016 • SQL 2017 Supported target servers: • SQL 2008 • SQL 2008 R2 • SQL 2012 • SQL 2014 • SQL 2016 • SQL 2017
Microsoft Azure	Source data: • Local backups of Microsoft Azure • Downloaded copies of database files • Downloaded copies of backup files Target Servers: • SQL Server data files in Microsoft Azure • SQL Server virtual machines in Microsoft Azure • SQL databases in Microsoft Azure
Virtual environments	ⓘ Note: Virtual operation of tape devices may have restrictions imposed by virtual operating systems.
Additional software	ItemPoint requires Microsoft .NET Framework version 4.5 ⓘ Note: In order for Dell EMC ItemPoint for Microsoft SQL Server to operate fully, you must ensure the Dell EMC software is correctly licensed for use and the source files are located on Dell

Table 33 ItemPoint for SQL Server requirements (continued)

Component	Requirement
	EMC storage. If not, attempts to open a source database will produce an error message.

Supported data types

ItemPoint supports the following data types:

- Table level restoring of data

- Snapshots (if presented as a drive letter or volume of files)

- Column (data) types (up to 2 GB):

  - BigInt, Int, SmallInt, TinyInt, Bit

  - Binary, VarBinary

  - Char, VarChar, NChar, NVarChar

  - Computed Columns

    (i) **Note:** Computed Columns appear in the Source List as <COMPUTED>. Computed Columns referencing an object that does not exist in the target database cannot be copied.

  - DateTime, SmallDateTime

  - Float, Real

  - Image, VarBinary(max)

  - Money, SmallMoney

  - Numeric, Decimal

  - Sql_variant

    (i) **Note:** If a sql_variant column is copied and:

      - The sql_variant is a Money or SmallMoney type, it is converted to a decimal on the target.

      - The value is a string type, the collation is not preserved.

  - Sparse columns without COLUMN_SET

  - Text, NText, VarChar(max), NVarChar(max)

  - Uniqueidentifier

  - XML

Limitations

The following features and capabilities are not supported:

- Database encryption and compression

- Backup data encryption

- SQL Server database objects other than tables, rows, and columns (such as system tables, stored procedures, triggers, primary and foreign keys, and CLR) are not supported and do not appear in the Dell EMC ItemPoint for Microsoft SQL Server interface.

- Limited support column (data) types:

  (i) **Note:** These column types appear in the Source List as either <UNSUPPORTED> or binary and are copied as binary.

- CLR UDT (User Defined Type)
- Cursor
- Date, DateTimeOffset, DateTime2, Time
- Geography, Geometry
- HierarchyId
- TimeStamp/RowVersion

- Unsupported column (data) types:
  - COLUMN_SET column
    (i) **Note:** While sparse columns defined by COLUMN_SET column appear in the Source List, tables containing this column cannot be copied.
  - Filestream
    (i) **Note:** Tables containing Filestream columns cannot be accessed.

- Unsupported backup types:
  - Files and filegroup
    (i) **Note:** Memory-Optimized tables cannot be accessed.
  - Columnstore indexes
    (i) **Note:** Tables with a columnstore index cannot be accessed.

Restoring Data Limitations

- Referential integrity, indexes, and permissions are not restored.

- When restoring tables with TimeStamp columns, the TimeStamp column is not preserved. The value is the time of the restoration for that row. This is the normal behavior for TimeStamp columns.

- Restoring tables with identity columns does not preserve the value of the identity column; it follows the rules of the identity column on the target table.
  For example:

  If the source table has three rows with identity values of 1, 3, 5, once restored, the target will have values of 1, 2, 3. (This is the normal behavior of identity columns.)

Installation requirements

To use the SQL GLR feature, you must select the following options from the NMM installation wizard:

- The SQL GLR option
  This option installs the following required components for GLR:

  - The Eldos CBFS driver
  - The NMM mount service
  - The NetWorker virtual file system (NWFS) plug-in
  - ItemPoint for SQL Server

- The SQL SSMS plug-in option
  This option is required because SQL GLR is managed and initiated through the Table Restore tab in the NMM Microsoft SQL Server Management Studio (SSMS) plug-in GUI.

The *NetWorker Module for Microsoft Installation Guide* provides more information.

Backup considerations

SQL GLR requires backups that are created with NMM 9.1 or later. No additional configuration is required for GLR compatible backups as GLR is enabled by default for backups that are taken with

NMM 9.1 or later. Backups that are taken with older versions of NMM are not GLR-enabled. Backups that use the NetWorker AES encryption or NetWorker compression technology are not supported for GLR.

GLR is supported only for SQL Server backups that are taken with an AFTD or Data Domain device. Client Direct must be enabled for the volumes containing the save sets to be restored in order for the mount operation to succeed. If a Client Direct session cannot be established, the mount operation fails.

If the backup is on a tape device, the backup must be cloned to an AFTD or Data Domain device before you can perform a GLR operation. In this scenario, it is also required to set the recovery pool in the registry. Contact support for details.

The default creation of SQL GLR compatible backups does not affect backup workflow. For example, if you are performing a type of backup that is not supported for GLR, the backup completes successfully. However, if a backup is not GLR-compatible, NMM writes warning messages to the operational logs. If these warning messages impact group completion statuses, disable GLR backups by typing `ENABLE_GLR=NO` in the application information field of the client resource.

The following backup types are not supported for SQL table-level restore:

- Columnstore indexes
  - (i) Note: Tables with a columnstore index cannot be accessed.

- File level and file group backups
  - (i) Note: Memory-Optimized tables cannot be accessed.

NMM SSMS plug-in considerations

ItemPoint for SQL Server requires administrative privileges, and as a result the NMM SSMS plug-in must be started with a user account that has administrative privileges.

The Table Restore tab may display backup versions that are not GLR-enabled. The mount operation fails if any of the save sets in the selected backup version are not GLR-capable. The VFS log file contains details of the cause of failure.

NMM allows only one backup mount to be active at a time. This restriction is built-in to the NMM SMSS plug-in GUI workflow. While a mount is active, the Table Restore tab is blocked for use until the ItemPoint for SQL Server GUI is closed.

Mount considerations

The maximum number of recovery sessions the NWFS process (nwfs.exe) uses while mounting a backup is 31 sessions. If NetWorker server parallelism is set and the value is less than 31, the NWFS process uses the parallelism value.

If the number of open recovery sessions causes any operational problems with the NetWorker Server, it is possible to reduce the number of recovery sessions by modifying the client parallelism registry setting. Contact support for details.

Large database considerations

SQL table level restore can be inappropriate for large databases for scenarios where a large amount of data has been committed to the database before to the backup.

The committed data is represented as outstanding transactions when there are SQL Server full and incremental backups on the database. When you perform table restores, ItemPoint must load all the outstanding transactions that are present in a backup. To load large databases, ItemPoint requires a few hours to read the data because of suboptimal Virtual File System performance.

This problem occurs while ItemPoint is loading the database, and ItemPoint may appear to stop responding while reading the backup.

If you observe this problem, an alternative to using SQL table-level restore is to restore the backup as flat files to a server where you have enough space, and the use ItemPoint with the files on the local disk.

Restore considerations

Consider the following ItemPoint for Microsoft SQL Server restore limitations:

- Referential integrity, indexes, and permissions are not restored.

- When restoring tables with TimeStamp columns, the TimeStamp column is not preserved. The value is the time of the restoration for that row. This is the normal behavior for TimeStamp columns.

- Restoring tables with identity columns does not preserve the value of the identity column. The restored table follows the rules of the identity column on the target table.
  For example, if the source table has three rows with identity values of 1, 3, 5, once restored, the target has values of 1, 2, 3 because this behavior is the normal behavior of identity columns.

Performing Granular Level Recovery

In NMM SSMS plug-in **NetWorker** window, from the **Table Restore** tab, you can restore SQL Server table-level data, that is, perform GLR.

About this task

(i) Note: The **Table Restore** tab is only visible in the NMM SSMS plug-in GUI if the SQL GLR option is selected during NMM installation.

You can use the **Table Restore** tab to mount the backup in the NetWorker virtual file system so that it is available for GLR by using the ItemPoint GUI.

Procedure

1. In the SQL Server Management Studio, select the SQL Server instance that you want to restore, and then open the **NetWorker** window.

2. In the **NetWorker** window, on the **Table Restore** tab, under **Select a page**, click **General**.

   The **General** page appears.
   The following figure shows the **NetWorker** window open to the **Table Restore** tab, on the **General** page.

Figure 30 NetWorker dialog box—Table Restore General page
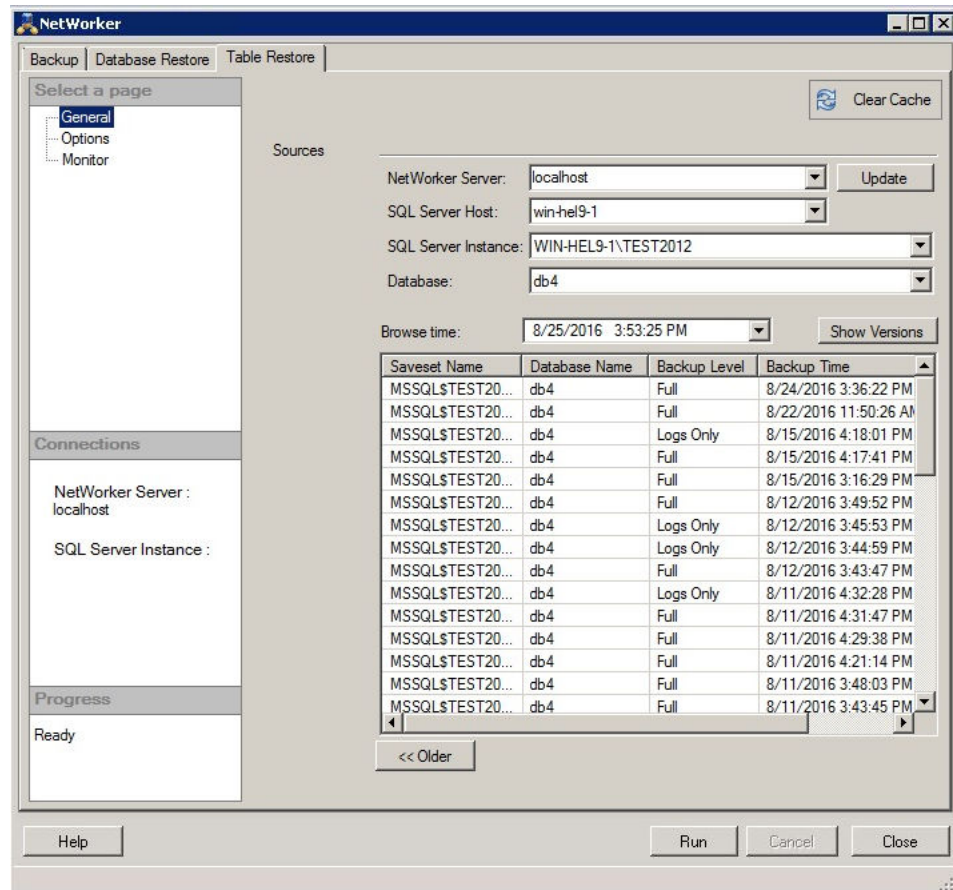
3. Select the NetWorker server where the backup is located from the **NetWorker Server** list.

   Use the **Update** button to update the list of available NetWorker servers.

   The **SQL Server Instance** list is refreshed after you select the NetWorker server.

4. From the **SQL Server Host** list, select the client that contains the backup.

   If you used a cluster or an availability group listener to perform the backup, select the same cluster or listener name from the **SQL Server Host** list.

5. From the **SQL Server Instance** list, select the server instance that contains the backup.

6. From the **Database** list, select the database.

7. Specify the browse time by using one of the following options:

   • Use the **Browse time** list.

   • Click **Show versions** and choose the browse date and time.

   The table is populated with the save sets in the SQL database instance that are available for restore. The number of available backup versions appears in the bottom bar.

8. From the table, select the save set that contains the items for GLR.

9. To specify additional GLR options, under **Select a page**, click **Options**.

   The **Options** page appears.

   The following figure shows the **NetWorker** window open to the **Table Restore** tab, on the **Options** page.

Figure 31 NetWorker dialog box—Table Restore Options page
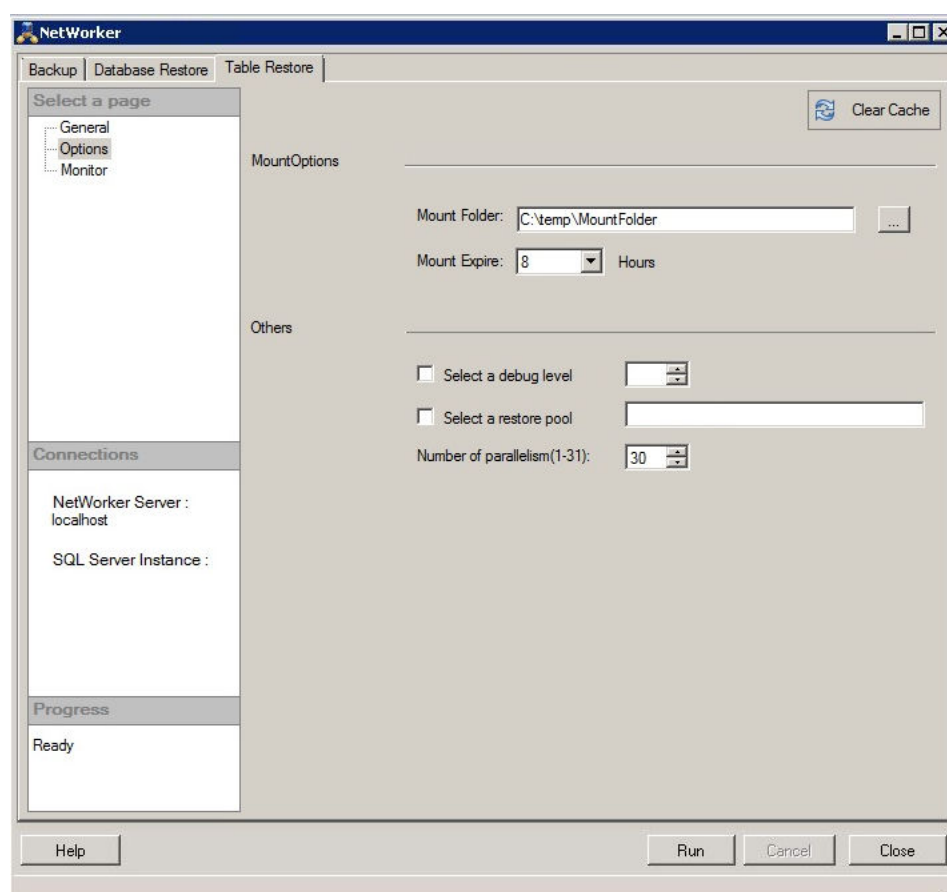
10. (Optional) Under **Mount Options**, modify the following settings as required:

    - Edit the **Mount Folder** field to change the location where database is mounted as files. The default mount path is the registry key folder that is specified during installation.

    - Edit the **Mount Expires** field to change the amount of time that the backup remains mounted. The default setting is 8 hours, and the maximum setting is 24 hours. Valid mount expiration values are 2, 4, 8, 16, and 24 hours.

11. (Optional) Under **Others**, modify the following settings as required:

    - To select the level of information recorded in the logs which can be used in troubleshooting restore issues, select **Select a debug level**. Levels range 0-9, with 0 representing no information and 9 representing the most amount of information.

    - To specify the clone pool where the save set is restored, select **Select a restore pool**, and then type the name of the restore pool in the field.

    - To change the restore parallelism setting, type or select a value in the **Number of parallelism (1-31)** field. Parallelism controls the number of restore sessions the NWFS process uses while mounting a backup. The default and maximum value is 31.

12. To mount the selected backup, click **Run**.

    After clicking **Run**, ItemPoint for Microsoft SQL Server opens and the NMM SSMS plug-in GUI switches to the **Monitor** page.
    The **Monitor** page provides messages about the status of the mount operation.

The following figure shows the **NetWorker** window open to the **Table Restore** tab, on the **Monitor** page.

Figure 32 NetWorker dialog box—Table Restore Monitor page
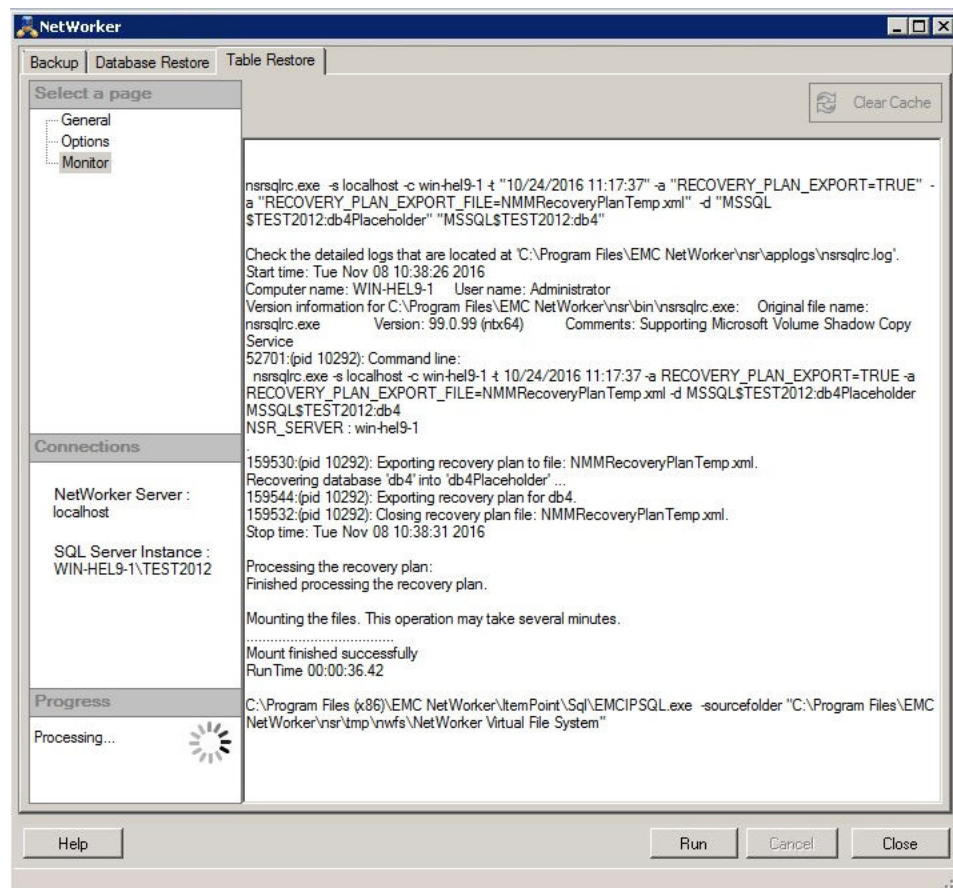
13. Perform GLR of the mounted database by using the ItemPoint for Microsoft SQL Server software.

    The *ItemPoint for Microsoft SQL Server User Guide* provides instructions about performing GLR. The NMM SSMS plug-in GUI is disabled till GLR is complete with ItemPoint.

Dismounting backups after performing GLR

After performing GLR using ItemPoint for Microsoft SQL Server, the backup must be dismounted from the virtual drive.

About this task

For standard restore operations, the backup is dismounted in the NMM SSMS plug-in GUI after the restore is completed and ItemPoint is closed.
If the NMM SSMS plug-in GUI cannot dismount the backup:

- You can manually dismount the backup by right-clicking the Mount system tray icon and clicking **Dismount Backups**.

- The backup is dismounted based on the timeout setting specified in the **Options** view. The default timeout period for a mounted backup is 8 hours.

Using the Mount system tray icon

About this task

The Mount system tray icon appears when a backup is mounted.

Click the Mount system tray icon to choose from a menu of available actions.

The following figure shows the Mount system tray menu options.

Figure 33 Mount system tray menu
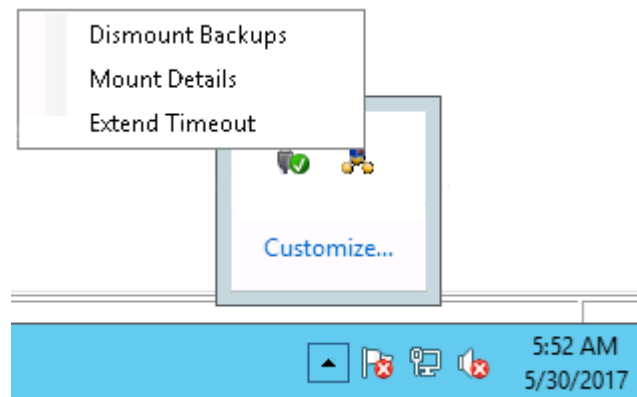
The following actions are available through the Mount system tray icon:

- **Dismount Backups**—Dismount a backup.

- **Mount Details**—View mount details.

- **Extend Timeout**—Extend the period for which the backup is mounted.
  The mount timeout can be extended by 2, 4, 8, 16, or 24 hours.

The following figure shows the **Mount Details** window.

Figure 34 Mount Details window
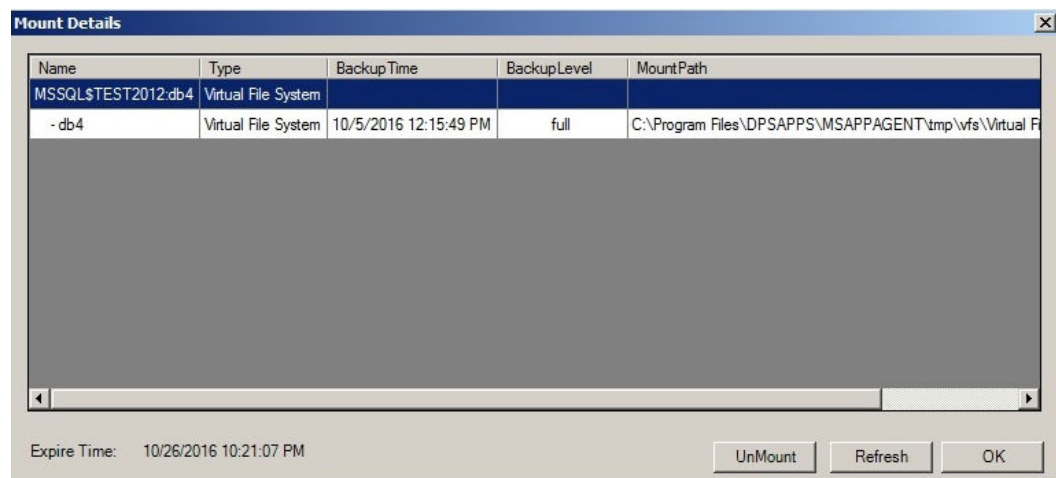
CHAPTER 8

Striped Backup and Recovery

This chapter includes the following sections:

Overview of striped backups

NMM supports the use of multiple stripes for backing up and restoring SQL Server data. Stripes are one or more data streams that are extracted in parallel from a database and written in parallel to multiple media devices, such as tape drives. With NMM, striping can yield a significant performance advantage when a large amount of data is backed up and recovered with multiple tape drives.

Although the SQL Server supports 64 stripes, NMM supports only 32 stripes. This restriction applies to the `nsrsqlsv` and `nsrsqlrc` commands.

When you perform striped backup and recovery operations, ensure that you assign the required user privileges. Read Access privileges for backup and recovery on page 44 for more information.

Before a striped backup begins, you must set the **Parallelism** field in the **Set Up Server** dialog box in NMC to at least one more than the number of stripes that the backup will use. For example, if you want to use three stripes, specify a value of four or more for the parallelism.

The total number of save streams (SS), which is the data and save set information that is being written to a storage volume during a backup, is the product of client parallelism (P) multiplied by the number of stripes (S), that is, SS = P X S.

NMM supports interleaving for backing up multiple clients to a single backup device. Interleaving writes multiple stripes to the same volume. Recovering interleaved backups can require substantial recovery time. When restoring an entire interleaved backup, one stripe is recovered at a time. This recovery method requires multiple passes over the same areas on the backup medium. When you recover data through an interleaved backup, recover each client's backup separately.

You can specify striping from the following locations:

- The NetWorker Administration GUI for a scheduled backup.

- The Windows client host command prompt.
  For example, type:

  `nsrsqlsv.exe -s ` ***`server_name`*** ` -c ` ***`client_name`*** ` -l full -S n ` **`"MSSQL:db_name"`**

  where:

  - Lowercase `-s` specifies the NetWorker server name.

  - Uppercase `-S` with `n` specifies the number of stripes.

Performance considerations

Review the following suggestions in this section to improve striped backup performance.

- In the NetWorker Administration GUI, in the **Create Device** or **Edit Device** dialog box, configure target sessions for the NetWorker devices that are performing the striped backup. Always specify one session per device when using striping. This configuration eliminates interleaving, yielding the best backup performance on the client host.

- Keep the following criteria in mind when you allot a separate backup device (such as a tape drive) for each strip in the backup or recovery operation:

  - Use a different NetWorker server for the backup and the recovery operations that is unrelated to the SQL Server.

  - Use a dedicated storage node for the striped backups.

- NMM automatically recovers data using the same number of stripes that were specified for the backup.

For best recovery performance, ensure that the same number of devices that are used for the backup are available at recovery time.

- Configure the computer to run the following components:

  ▪ NetWorker client as a NetWorker storage node

  ▪ NMM

  ▪ SQL Server

ⓘ **Note:** The NetWorker server cannot be installed on the same host as the NMM client software.

Performing striped backups

NMM supports manual and scheduled backup striping.

About this task

You must complete the required steps to perform a manual striped backup from the SQL Server host, which is a NetWorker client.

- From the NMM Microsoft SQL Server Management Studio (SSMS) plug-in GUI:

  1. Open the NMM SSMS plug-in GUI.

  2. Select the **Backup** tab.

  3. Under **Select a page**, click **General**.

  4. Select the NetWorker server.

  5. Select the SQL Server instance.

  6. Select the database filter.

  7. Select an item to back up.

  8. Select the backup type.

  9. In **Backup** > **Options**, select **Create a Striped Backup**.

  10. In the **Stripes** list, select the number of stripes.

- From the Windows command prompt, specify the –S option with the nsrsqlsv command. For example, at the command prompt, type:

  ```
  nsrsqlsv.exe -s server_name -c client_name -l full -S n "MSSQL:db_name"
  ```

  where *n* specifies the number of stripes you want to use, for example, -S*3*.

- From the NetWorker Administration GUI:

  1. Click **Protection**.

  2. In the expanded left pane, click **Clients**.

  3. In the right pane, right-click the client, and then select **Modify Client Properties**.

  4. On the **Apps & Modules** tab, append –S*n* to the nsrsqlsv command, where *n* is the number of stripes to use.

  5. Click **OK**.

Performing striped recovery

NMM performs striped recovery optimizations so striped recoveries proceed as quickly as possible.

Parallelism is the number of simultaneous data streams during a restore operation. Changing parallelism affects the volume of data that host can process at the same time to reduce restore times and optimize performance.

NMM distinguishes between disk type and tape type devices when determining parallelism. For disk type devices, NMM performs recovery operations with as many parallel streams as the NetWorker Server allows. For tape type devices, the parallelism value is limited to the number of tape devices where each stripe resides.

Refer to the *NetWorker Administration Guide* for details on the device parallelism that can be specified on the NetWorker Server.

Optimal striped recovery operations

Striped recovery assumes, by default, that enough NetWorker tape sessions are available to enable the striped recovery to proceed without blocking.

NMM then assumes the following:

- For tape devices, each stripe was written to a different volume during backup. Interleaving was not used during the striped backup.

- The same number of devices are available during both backup and restore operations. No device has failed since the backup.

- All devices that are used for backup are available. No other client sessions are assigned to the devices.

If the configuration meets all of these requirements, NMM restores the striped recovery. Backup data is delivered to the SQL Server when each tape session is established. This method provides the highest performance and lowest restore time.

Fail-safe striped recovery operations

If a striped backup is interleaved or if fewer tape sessions are available at the time of recovery than during backup, because of a failed or busy device, a fail-safe striped recovery is necessary.

NMM contacts the NetWorker server with the list of striped save sets to determine the available number of tape sessions before it starts striped recovery. If the number of tape sessions is smaller than the number of striped save sets, NMM uses a fail-safe striped recovery algorithm. The algorithm that NMM selects depends on the SQL Server version.

The following table provides guidelines for determining when to activate the fail-safe striped recovery algorithm.

Table 34 Guidelines for fail-safe striped recovery

If	Then
The backup was interleaved	Use NMC to determine if a second mount request has occurred for a volume that is already assigned tape sessions.

Table 34 Guidelines for fail-safe striped recovery (continued)

If	Then
Fewer devices are available	Use NMC to determine if a device has been deactivated or is performing an operation for another client.

About fail-safe striped recovery

When you enable a fail-safe striped recovery for a SQL Server client, additional configuration settings are recommended so the recovery can proceed at maximum throughput.

NMM takes advantage of the SQL Server VDI removable pipes feature. This feature allows third-party backup vendor products to recover a striped backup from fewer devices. However, NMM cannot accurately determine how many tape sessions the NetWorker server can assign. The NetWorker server does not support striped recoveries by default. Therefore, only one tape session is available to recover the striped save sets before the detection phase.

The SQL Server VDI removable pipes function allows a third-party backup vendor to start as many stripe recovery threads as there are tape sessions available. Otherwise, the recovery stops responding.

ⓘ **NOTICE** Because enabling striped recoveries on the NetWorker server might cause file system recoveries to fail, do not use this recovery method as a permanent solution.

Performing fail-safe striped restore

The procedure in this section applies to normal and copy restore types.

About this task

You must complete the required steps to enable a fail-safe striped restore.

Procedure

1. Open the NMM Microsoft SQL Server Management Studio (SSMS) plug-in GUI.

2. On the **Database Restore** tab, under **Select a page**, click **General**.

3. Under **Sources**, make the following selections:

   • Select the NetWorker server.

   • Select the SQL Server host.

   • Select the SQL Server instance.

   • In the list of backed-up records that appears, select the item to restore.

4. To specify additional restore options, under **Select a page**, click **Options**.

   The **Options** page appears.

5. Under **Restore Options**, select the restore options.

6. (Optional) Enable striped restores on the NetWorker server by creating the file `striped_recovery` in the `\nsr\debug` directory on the NetWorker server.

   ⓘ **NOTICE** You do not need to restart the NetWorker services to activate this setting.

7. From the NMM SSMS plug-in GUI, start the restore operation.

8. If the volumes with the striped restore are not managed by an autochanger, monitor events in NMC.

When a media wait event occurs, load the volume.

CHAPTER 9

Disaster Recovery

This chapter includes the following sections:

Overview of disaster recovery

The examples that follow provide general principles and procedures for recovering data.

Before beginning a SQL Server disaster recovery, review the following:

- *NetWorker Server Disaster Recovery and Availability Best Practices Guide*

- *NetWorker Release Notes*

- Microsoft SQL Server Books Online

To rebuild system databases for SQL Servers with the command prompt, refer to the information provided at the following link: http://msdn.microsoft.com/en-us/library/ms144259(v=sql.100).aspx

(i) Note: The SQL Server databases must be recovered in the correct order. Follow the database recovery steps in the Data Recovery chapter.

Disaster recovery features

The following table lists the disaster recovery features that NMM supports.

Table 35 Disaster recovery features

Disaster recovery features	Descriptions
System database restore automation	Certain system databases require SQL Server service control, including the master and the `msdb` databases. NMM automates the control of these services as follows: • For the master database, the SQL Server restarts in single-user mode as required by the SQL Server. • For the `msdb` database, the SQL Agent shuts down to close the connections to the `msdb` database. (i) Note: NMM does not support snapshot backups or restores of the SQL Server master database. Restore from a traditional backup of the master system database during a disaster recovery.
Database restore order	When you restore a complete backup of all databases or when you restore certain system databases, the restore must occur in a specific order. If system databases are present in the restore list, NMM ensures that the restore order follows SQL Server procedures as follows: 1. The master database is always restored first. This order ensures that metadata present in the master database is correct for all subsequent restored databases. 2. The `msdb` database is always restored after the master database and before all other databases. This sequence ensures that scheduling and other system data present in the `msdb` database are correct for all subsequent restored databases. 3. The model database is always restored after the master and the `msdb` databases and before all other databases. This sequence ensures that the database configuration is correct for all subsequent databases.

Table 35 Disaster recovery features (continued)

Disaster recovery features	Descriptions
SQL Server startup complete detection	When the SQL Server is started, a database startup process is also started. The SQL Server enables user connections while the startup process is running. However, if the startup process is interrupted by a database `RESTORE` query, any database that has not yet started is marked as suspect. When the interruption occurs, subsequent restores of the `msdb` database fail. Subsequent restore of any user database might also fail unless the **Overwrite the Existing Database** field is specified in the **Properties** dialog box. NMM waits for the SQL Server to complete the database startup process for all databases before starting a restore. NMM waits for the restore of the master database to complete. This wait process enables a proper restore of the `msdb` database and user databases following a SQL Server startup. Always use the `-f` option with the `nsrsqlrc` command for a restore that follows a SQL Server startup.
Overwriting capability	Use the **Files/Filegroups** page settings on the **Database Restore** tab in the NMM SSMS plug-in GUI to specify that you want to overwrite all databases during a restore operation. NMM does not handle all dependent services. When restoring application services, such Metadata or Replication services, as well as the databases on which these services depend, the services must be manually shut down. NMM does not ensure that all connections to a database are closed before it restores a database as required by the SQL Server. Such open connections must be manually closed. The Microsoft SQL Server documentation provides more information.

Performing disaster recovery

If NMM binary files or SQL Server binary files are damaged or lost, follow the instructions that are provided in this section.

The *NetWorker Server Disaster Recovery and Availability Best Practices Guide* provides more information about using NetWorker for disaster recovery.

When not to reinstall the SQL Server

During a disaster recovery of a SQL Server, you must not install the SQL Server in certain cases.

Do not reinstall the SQL Server if any of the following conditions exist:

- A recent NetWorker backup of the ALL save set exists.
- The SQL Server was installed before the ALL save set backup was performed.
- The SQL Server backups for all of the SQL Server databases exist.

If a SQL Server is installed and then the ALL backup is performed, critical state information for the SQL Server is backed up as part of the ALL save set. When the ALL save set contains the critical state information for the SQL Server and backups for all of the SQL Server databases exist, you can recover the SQL Server by restoring the ALL save set and the SQL Server databases.

Restoring a damaged primary disk

Procedure

1. Shut down the SQL Server.

2. Rebuild the databases on the SQL Server. Information about rebuilding various SQL Servers is available on the Microsoft MSDN website.

   The following table provides links to relevant Microsoft documentation for various versions of SQL Server.

   Table 36 Rebuilding SQL Server system databases

SQL Server version	Microsoft article
SQL Server 2017 and 2016	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.130).aspx
SQL Server 2014	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.120).aspx
SQL Server 2012	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.110).aspx

3. Restart the SQL Server.

4. In the NetWorker User for SQL Server GUI, select **Replace** to restore the SQL Server master database and the `msdb` database.

5. Restore the other SQL application databases.

   (i) Note: To run steps 4 and 5 as a single operation, from a command prompt, type `nsrsqlrc -s... -f MSSQL:`

6. Complete the disaster recovery through the NMM SSMS plug-in GUI:

   a. Open the NMM SSMS plug-in GUI.

   b. On the **Database Restore** tab, under **Select a page**, click **General**.

   c. Under **Sources**, make the following selections:

      - Select the NetWorker server.
      - Select the SQL Server host.
      - Select the SQL Server instance.
      - Change the backup time, which is the current time by default.
      - In the list of backed-up records that appears, select all the items to restore.

   d. To specify additional recovery options, under **Select a page**, click **Options**.

      The **Options** page appears.

   e. Under **Restore Options**, select **Overwrite the existing database (WITH REPLACE)**.

   f. Click **Run**.

Restoring a damaged binary disk

Procedure

1. Shut down the SQL Server.

2. Restore the `ALL` save set.

   > (i) **Note:** When the `ALL` save set contains the critical state information for the SQL Server and backups for all of the SQL Server databases exist, you can recover the SQL Server by restoring the `ALL` save set and the SQL Server databases.

3. Restart the computer.

4. Shut down the SQL Server.

5. Rebuild the databases on the SQL Server. Information about rebuilding various SQL Servers is available on the Microsoft MSDN website.

   The following table provides links to relevant Microsoft documentation for various versions of SQL Server.

   Table 37 Rebuilding SQL Server system databases

SQL Server version	Microsoft article
SQL Server 2017 and 2016	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.130).aspx
SQL Server 2014	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.120).aspx
SQL Server 2012	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.110).aspx

6. Start the restore in the NetWorker User for SQL Server GUI. Select **Replace** to restore the SQL Server master database and the `msdb` database.

7. Restore the other SQL application databases.

   > (i) **Note:** You can run steps 6 and 7 as a single operation:
   > From a command prompt, type the following command:
   > `nsrsqlrc -s... -f MSSQL:`

8. Complete the disaster recovery through the NMM SSMS plug-in GUI:

   a. Open the NMM SSMS plug-in GUI.

   b. On the **Database Restore** tab, under **Select a page**, click **General**.

   c. Under **Sources**, make the following selections:

   - Select the NetWorker server.
   - Select the SQL Server host.
   - Select the SQL Server instance.
   - Change the backup time, which is the current time by default.
   - In the list of backed-up records that appears, select all the items to restore.

   d. To specify additional recovery options, under **Select a page**, click **Options**.

   The **Options** page appears.

> e. Under **Restore Options**, select **Overwrite the existing database (WITH REPLACE)**.
>
> f. Click **Run**.

Restoring SQL Server and NetWorker server

Complete the required steps if the disks with the SQL Server binaries and the NetWorker online indexes (the `nsr` file system) or the SQL Server data are damaged.

Procedure

1. Recover the NetWorker binaries and online indexes. The *NetWorker Server Disaster Recovery and Availability Best Practices Guide* provides details.

2. Reinstall NetWorker:

   - If the computer that failed is the NetWorker server, reinstall the NetWorker server.

   - If the computer that failed is a NetWorker client, reinstall the NetWorker client.

3. Recover the `ALL` save set.

   (i) Note: When the `ALL` save set contains the critical state information for the SQL Server and backups for all of the SQL Server databases exist, you can recover the SQL Server by restoring the `ALL` save set and the SQL Server databases.

4. Restart the computer.

5. Shut down the SQL Server services if they are running.

6. Rebuild the databases on the SQL Server. Information about rebuilding various SQL Servers is available on the Microsoft MSDN website.

   The following table provides links to relevant Microsoft documentation for various versions of SQL Server.

   Table 38 Rebuilding SQL Server system databases

SQL Server version	Microsoft article
SQL Server 2017 and 2016	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.130).aspx
SQL Server 2014	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.120).aspx
SQL Server 2012	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.110).aspx

7. Restart the SQL Server.

8. In the NetWorker User for SQL Server GUI, select **Replace** to restore the SQL Server master database and the `msdb` database.

9. Restore the other SQL application databases.

   (i) Note: You can run steps 8 and 9 as a single operation:
   From a command prompt, type the following command:
   `nsrsqlrc -s... -f MSSQL:`

10. Complete the disaster recovery through the NMM SSMS plug-in GUI:

    a. Open the NMM SSMS plug-in GUI.

    b. On the **Database Restore** tab, under **Select a page**, click **General**.

    c. Under **Sources**, make the following selections:

- Select the NetWorker server.

- Select the SQL Server host.

- Select the SQL Server instance.

- Change the backup time, which is the current time by default.

- In the list of backed-up records that appears, select all the items to restore.

    d. To specify additional recovery options, under **Select a page**, click **Options**.

       The **Options** page appears.

    e. Under **Restore Options**, select **Overwrite the existing database (WITH REPLACE)**.

    f. Click **Run**.

Restoring the SQL Server without reinstallation

About this task

If a SQL Server is installed and then the `ALL` backup is performed, critical state information for the SQL Server is backed up as part of the `ALL` save set. When the `ALL` save set contains the critical state information for the SQL Server and backups for all of the SQL Server databases exist, you can recover the SQL Server by restoring the `ALL` save set and the SQL Server databases.

Procedure

1. Reinstall the operating system.

2. Reinstall NetWorker:

   - If the computer that failed is the NetWorker server, reinstall the NetWorker server.

   - If the computer that failed is a NetWorker client, reinstall the NetWorker client.

3. Restore the `ALL` save set.

4. Restart the computer.

5. Shut down the SQL Server.

6. Rebuild the databases on the SQL Server. Information about rebuilding various SQL Servers is available on the Microsoft MSDN website.

   The following table provides links to relevant Microsoft documentation for various versions of SQL Server.

   Table 39 Rebuilding SQL Server system databases

SQL Server version	Microsoft article
SQL Server 2017 and 2016	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.130).aspx
SQL Server 2014	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.120).aspx
SQL Server 2012	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.110).aspx

7. Restart the SQL Server.

8. In the NetWorker User for SQL Server GUI, select **Replace** to restore the SQL Server master database and the `msdb` database.

9. Restore the other SQL application databases. Restore the SQL Server to the most recent backup.

> (i) **Note:** To run steps 8 and 9 as a single operation from the command line, type the following command:
>
> ```
> nsrsqlrc -s... -f MSSQL:
> ```
>
> For information about how you can install various SQL Server versions from the command prompt, visit http://msdn.microsoft.com/en-us/library/ms144259(v=sql.100).aspx. Use the Other Versions list at the top of the page to view information for a particular SQL version.

10. Complete the disaster recovery through the NMM SSMS plug-in GUI:

    a. Open the NMM SSMS plug-in GUI.

    b. On the **Database Restore** tab, under **Select a page**, click **General**.

    c. Under **Sources**, make the following selections:

       • Select the NetWorker server.

       • Select the SQL Server host.

       • Select the SQL Server instance.

       • Change the backup time, which is the current time by default.

       • In the list of backed-up records that appears, select all the items to restore.

    d. To specify additional recovery options, under **Select a page**, click **Options**.

       The **Options** page appears.

    e. Under **Restore Options**, select **Overwrite the existing database (WITH REPLACE)**.

    f. Click **Run**.

Restoring the SQL Server

About this task

Use the following procedure to restore the SQL Server.

Procedure

1. Reinstall the operating system.

2. Reinstall NetWorker:

   • If the computer that failed is the NetWorker server, reinstall the NetWorker server.

   • If the computer that failed is a NetWorker client, reinstall the NetWorker client.

3. Restore or reinstall the SQL Server software.

   Do not reinstall the SQL Server if any of the following conditions apply:

   • A recent NetWorker backup of the ALL save set exists.

   • The SQL Server was installed before the ALL save set backup was performed.

   • The SQL Server backups for all of the SQL Server databases exist.

4. Start the SQL Server.

5. When restore is started, in the NetWorker User for SQL Server GUI, select **Replace** to restore the SQL Server master database and the `msdb` database.

6. Restore the other SQL application databases.

   (i) Note: To run steps 5 and 6 as a single operation from the command line, type the following command:

   ```
   nsrsqlrc -s... -f MSSQL:
   ```

7. Complete the disaster recovery through the NMM SSMS plug-in GUI:

   a. Open the NMM SSMS plug-in GUI.

   b. On the **Database Restore** tab, under **Select a page**, click **General**.

   c. Under **Sources**, make the following selections:

   - Select the NetWorker server.
   - Select the SQL Server host.
   - Select the SQL Server instance.
   - Change the backup time, which is the current time by default.
   - In the list of backed-up records that appears, select all the items to restore.

   d. To specify additional recovery options, under **Select a page**, click **Options**.

   The **Options** page appears.

   e. Under **Restore Options**, select **Overwrite the existing database (WITH REPLACE)**.

   f. Click **Run**.

CHAPTER 10

Bare Metal Recovery

This chapter includes the following sections:

Planning bare-metal recovery

This section provides an overview of the Windows bare-metal recovery (BMR) solution and considerations when planning a BMR.

Overview

Bare-metal recovery (BMR) is a technique in the field of data recovery and restoration where the backed up data is available in a form that allows you to restore a system from bare metal, that is, without any requirements as to previously installed software or operating system.

Typically, the backed up data includes the necessary operating system, applications, and data components to rebuild or restore the backed up system to an entirely separate piece of hardware. The hardware receiving the restore should have a similar configuration as that of the hardware that was the source of the backup.

The basic BMR is the process of bringing up a server after a disaster and ensuring that the system recovers with the operating system, the applications, and the data as they were at the time of the failure.

Restoring a server to the exact configuration that it had at the time of its destruction can be a difficult task. When this restoration is performed to another hardware, it can add to the complexity of the process and can be time-consuming. Windows BMR solution provides a flexible and reliable method of restoring a server after a disaster.

System requirements

The following sections list requirements to perform Windows BMR. However, the *NetWorker E-LAB Navigator* , which is available at https://elabnavigator.emc.com/eln/elnhome, provides the latest information about the system requirements to perform Windows BMR by using NMM.

Supported operating systems

The *NetWorker E-LAB Navigator* , which is available at https://elabnavigator.emc.com/eln/elnhome, provides information about the operating systems and versions that NMM supports.

Supported Microsoft applications

The *NetWorker E-LAB Navigator* , which is available at https://elabnavigator.emc.com/eln/elnhome, provides information about the Microsoft applications and versions that NMM supports.

NetWorker software version requirements

The *NetWorker E-LAB Navigator* , which is available at https://elabnavigator.emc.com/eln/elnhome, provides information about the NetWorker software versions that NMM supports.

Use the NetWorker Windows BMR 32-bit ISO image to recover an x86 operating system on either an x86 or x64 computer.

Use the NetWorker Windows BMR 64-bit ISO image to recover only an x64 operating system on an x64 computer.

Microsoft BMR requirements

Perform BMR recoveries to the same or similar hardware, and physical to virtual environment.

The following Microsoft KB article provides the requirements to perform a BMR to similar hardware:

http://support.microsoft.com/kb/249694

CPU requirements

Consider the following CPU requirements:

- The operating system architecture and the processor architecture must match.

- Basic Input/Output System (BIOS) or Unified Extensible Firmware Interface (UEFI) must match.

- You can treat AMD or Intel processors as being the same if they follow the same architecture. You can recover the operating system backup of an AMD x64 computer to an Intel x64 computer. The process is reversible.

- You can restore the backup of an x86 operating system version only to an x86 processor computer.

- You can restore the backup of an x64 operating system version only to an x64 processor computer.

Hard disk requirements

Consider the following hard disk requirements:

- The disk or RAID drivers that are used in the old system must be compatible with the disk or RAID controllers in the new system.

- For each critical disk on the BMR target system, the startup hard disk capacity on the new system must be greater than or equal to the capacity on the old system. BMR fails if the capacity is smaller by even a single byte.

- Windows BMR supports IDE, SATA, or SCSI hard disks. You can perform the backup on one type of hard disk and recover on another type. For example, SAS to SATA is supported.

- You must restore the backup to the same logical disk numbers as on the original server. You cannot use different logical disk numbers on the target system to recover the critical volumes such as the operating system volume.

- Ensure that the RAID setup on the destination computer does not interfere with the disk order of the hard disks.

NIC driver requirements

To use different Network Interface Card (NIC) after a Windows BMR recovery, install new NIC drivers that match the NIC in the new computer after Windows starts on the new server.

Critical and noncritical volume requirements

Consider the following critical and noncritical volume requirements:

- Windows BMR backs up only critical volumes, and can be used for offline disaster recovery. Use the NetWorker client to back up the non-critical volumes.

  (i) Note: NetWorker considers only system volume as a critical volume. If you have installed a Microsoft application on a drive other than the system drive, the drive is not considered as critical. On Windows Server 2008 R2, a volume is critical if a Microsoft application has installed a Windows service on it, but on Windows Server 2012, a volume that has a Windows application service installed is not critical.

- To make a volume critical on Windows Server 2012, set the value of the `HKLM\System\CurrentControlSet\Control\SystemWriter\ReportWin32ServicesNonSystemState` registry key to 0. This ensures that BMR

includes the Microsoft application binaries, and the volume on which they are installed is marked as critical.

(i) **Note:** This registry key may not be in the registry by default. If it is not in the registry, **create the** `HKLM\System\CurrentControlSet\Control\SystemWriter \ReportWin32ServicesNonSystemState` **registry key.**

System Reserved Partition requirements

Ensure that System Reserved Partition (SRP) is online before you perform a BMR backup. Otherwise, the backup fails and displays the following error messages:

```
84687:save: Unable to get volume information of file system '\\?
\Volume{245204f6-5ff7-11e2-a3ac-806e6f6e6963}\': The device is not
ready. (Win32 error 0x15). VSS OTHER: ERROR: VSS failed to process
snapshot: The shadow copy provider had an unexpected error while trying
to process the specified operation. (VSS error 0x8004230f)
```

```
90108:save: Unable to save the SYSTEM STATE save sets: cannot create the
snapshot.
```

```
86024:save: Error occured while saving disaster recovery save sets.
```

If SRP is offline, perform the following steps to bring it online:

1. In the WinPE command prompt, type **diskpart** and press **Enter**.

2. Run the following command to display the list of volumes:
   DISKPART> list volume

3. Run the following command to select the volume that is offline:
   DISKPART> select <*volume_name*>

4. Run the following command to bring the selected volume online:
   DISKPART> online volume

Protecting an environment before a disaster

To prepare for disaster recovery, back up application data and other necessary files.

Procedure

1. Use NetWorker client to back up non-application data.

   The *NetWorker Administration Guide* provides details about using the NetWorker client to backup non-application files.

2. If you use a NetWorker server earlier than 9.0.x, create a NetWorker group without enabling the Snapshot option. Otherwise, create a policy.

3. Create a NetWorker client resource, and assign it to the group that you created in step 2.

4. Configure the NetWorker client resource by typing **ALL** in the **Save set** field, and clearing the **Backup command** and **Application information** fields.

5. Perform a backup.

6. Use NMM to perform a full backup of application data.

   The NMM application specific user guides provide details about how to perform a full backup.

BMR by using NetWorker and NMM

Perform Windows file system backup and recovery by using the NetWorker client, and application-specific backup and recovery by using NMM.

(i) **Note:** Specific information about how to backup and recover Microsoft applications in NMM 19.2 is provided in application-specific user guides.

NetWorker disaster recovery provides an automated BMR solution by using the Windows ASR writer and other VSS writers to identify critical volumes that are required to perform a disaster recovery on a disabled system. BMR is performed offline, that is, when the Windows operating system is inactive. This avoids the necessity to manually reinstall Windows, and the problems that occur when operating system files are being restored to an active Windows system.

NMM 19.2 is compatible with the NetWorker 19.2 client, which provides a true BMR capability for Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, and Windows Server 2008. This capability is built into the NetWorker client and provides BMR support to the same or similar hardware. For Windows BMR, you should download the ISO recovery boot image file from the Online Support website (https://support.emc.com). The ISO image provides a wizard that allows you to format disks, recover Windows critical volumes from backup, and restart the server to bring it back online.

To support a NetWorker Windows BMR recovery, download the Windows BMR image from the Online Support website (https://support.emc.com). This download enables you to create a bootable Windows BMR ISO that contains NetWorker binaries and a wizard, which controls the recovery process.

The "Windows Bare Metal Recovery to Physical or Virtual Computers" section in the *NetWorker Administration Guide* provides detailed information about how to use the NetWorker Windows BMR image to perform a BMR recovery on protected hosts and VMware virtual machines.

(i) **NOTICE** For all the Microsoft applications, after performing Windows disaster recovery and restarting the system, check all the disk and volume configurations. Usually, the disks and volumes appear as on the original system.
However, it is possible, especially in BMR scenarios, that the volume or disk signatures do not match the original ones, and the non-critical volumes or disks are offline and not mounted. Use the Microsoft Disk Manager to bring the volumes and disks online, and then restart the system for drive letter reassignments. Assign the same drive letters that existed before the BMR. Non-critical volumes that the mount points access might have a similar issue.

SQL Server in a cluster environment

This section provides the bare-metal recovery (BMR) backup and recovery procedures for SQL Server by using the VSS technology in a cluster environment.

Backing up a SQL Server for BMR

Configure a SQL Server client resource to back up SQL Server for BMR.

Creating client resources by using the NetWorker Client Backup Configuration Wizard

Create separate client resources to back up the application data and the file system data. The client resource for the file system data must include the `ALL` save set.

About this task

The section Access privileges for backup and recovery on page 44 provides information about how to create a client resource to back up the application data.

The "Creating a client resource with the Client Backup Configuration Wizard" section in the *NetWorker Administration Guide* provides information about how to create a client resource to back up the file system data.

Creating client resources by using the Client Properties dialog box

Create separate client resources to backup the application data and the file system data. The client resource for the file system data must include the `ALL` save set.

About this task

This procedure uses the following sample SQL Server configuration:

- You have installed SQL Server on cluster volume `P:\`.

- The cluster quorum drive is `E:\`.

- You have clustered all the drives except the `C:\` drive in the cluster.

Procedure

1. In the Administration view of NMC, create a policy, a workflow, a group, and an action for SQL Server.

   The "Data Protection Policies" chapter in the *NetWorker Administration Guide* provides information.

2. Create a client resource to back up the `ALL` save set on the active node:

   a. In the **NetWorker Administrator** window, click **Protection**.

   b. In the expanded left panel, right-click **Clients** and select **New**.

   c. In the **Create Client** dialog box:

      a. In the **Name** field, type the name for the active node.

      b. In the **Group** list, select the group that you created in step 1.

      c. In the **Save set** field, type **ALL**.

      d. Specify other fields according to the requirements.

      e. Click **OK**.

3. Perform the backup.

   Ensure that the backup successfully completes.

4. Perform steps 2 and 3 for the passive node.

   (i) Note: When you perform step 2ca, in the **Name** field, type the name for the passive node.

5. Perform steps 2 and 3 for the quorum drive.

   (i) Note: When you perform step 2ca, in the **Name** field, type the name for the Windows cluster.

6. Perform steps 2 and 3 for the SQL cluster drives.

   (i) Note: When you perform step 2ca, in the **Name** field, type the SQL virtual name.

7. Create a client resource with SQL virtual name to back up the SQL application data:

   a. In the **NetWorker Administrator** window, click **Protection**.

b. In the expanded left panel, right-click **Clients** and select **New**.

The **Create Client** dialog box appears.

c. On the **General** tab:

a. In the **Name** field, type the SQL virtual name.

b. In the **Group** list, select the group that you created in step 1.

c. In the **Save set** field, type `APPLICATIONS:\SqlServerWriter`.

d. On the **Apps & Modules** tab, in the **Backup command** field, type `nsrnmmsv`.

e. Specify other fields according to the requirements.

f. Click **OK**.

8. Create dummy client resources for the physical nodes of the cluster.

9. Perform the backup.

Ensure that the backup successfully completes.

Performing BMR of a SQL Server cluster

The BMR process recovers `C:` drive only. It does not recover the SQL data, the quorum and SQL cluster drives, and the system databases such as master, msdb, and model.

Procedure

1. Separately recover the active and passive nodes by performing the procedures that the "Performing a Windows BMR recovery to physical or virtual computers" section in the *NetWorker Administration Guide* describes.

2. Recover the cluster quorum database:

a. After you have started the active and passive nodes, ensure that the cluster disks are online on the active node in the Disk Management Console of the Windows GUI.

b. Add the quorum drive to Failover Cluster Management of the Cluster Configuration UI.

c. Select the storage node in Failover Cluster Management and ensure that the quorum disk is online.

d. On the SQL Server active node, start the NetWorker User program.

e. Click **Recover**.

f. In the **Source Client** dialog box, select the source client with the Windows cluster name, and click **OK**.

g. In the **Destination Client** dialog box, select the destination client with the active node name, and click **OK**.

h. Select the quorum drive and recover the drive.

3. Recover the SQL cluster drives:

a. Add the SQL drives to Failover Cluster Management of the Cluster Configuration UI.

b. Ensure that the disk is online.

c. Configure the drive mount points between the drives so that the mount points are the same as the mount points during the backup.

d. On the SQL Server active node, start the NetWorker User program.

      e. Click **Recover**.

      f. In the **Source Client** dialog box, select the source client with the SQL virtual name and click **OK**.

      g. In the **Destination Client** dialog box, select the destination client with the active node name and click **OK**.

      h. Select all the SQL cluster drives and recover the drives.

4. Build the msdb and model databases:

      a. Start the physical nodes of the cluster.

      b. Ensure that you have the SQL Server installation software on the active node.

      c. In the software, go to the folder where the SETUP file exists and run the following command:

```
Setup/QUIET/ACTION=REBUILDDATABASE/INSTANCENAME=SQL Server instance
name/SQLSYSADMINACCOUNTS=Domain\administrator username/
IAcceptSQLServerLicenseTerms
```

Refer to Microsoft documentation for information about how to rebuild the SQL Server system databases. The following table provides links to relevant Microsoft documentation for various versions of SQL Server.

Table 40 Rebuilding SQL Server system databases

SQL Server version	Microsoft article
SQL Server 2017 and 2016	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.130).aspx
SQL Server 2014	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.120).aspx
SQL Server 2012	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.110).aspx

5. In the **Failover Cluster Management** window of the Cluster Configuration UI, perform the following steps:

      a. Open the **SQL Server Properties** dialog box.

      The following figure shows the **SQL Server Properties** dialog box.

Figure 35 SQL Server Properties dialog box
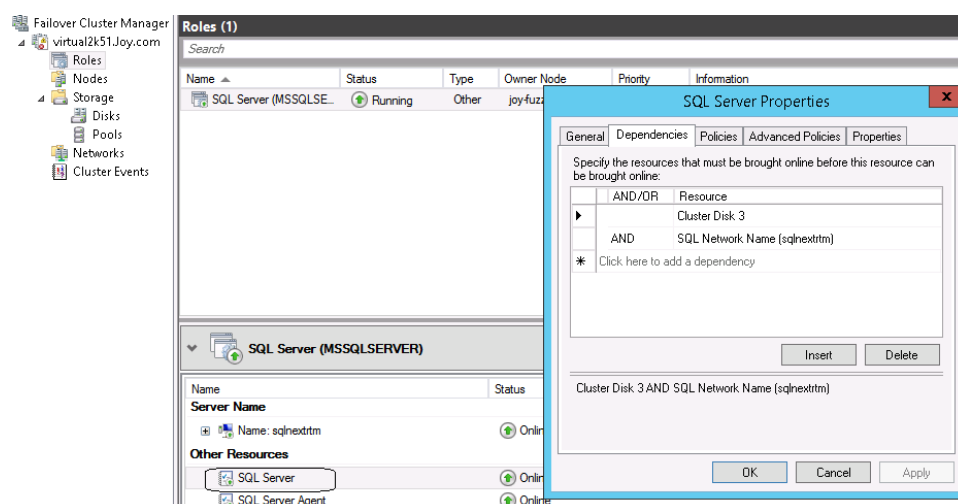
b. On the **Dependencies** tab, add all the SQL Server dependencies that you must bring online before you bring the SQL Server online. For example, add any dependent cluster disk.

c. Bring the SQL virtual resources online.

d. Bring the SQL Server online.

e. Perform the failover and ensure that the failover is successful.

6. Restore the databases by using the NetWorker User for Microsoft GUI:

a. Open the NetWorker User for Microsoft GUI.

b. Select the SQL Server and relevant SQL virtual server instance, and then click **Recover**.

c. On the **Recover** tab page, click the **Browse** tab.

d. Expand **APPLICATIONS**, and then select **SqlServerWriter**.

e. Perform the restore operation.

f. After the restore operation completes, ensure that the SQL Server service starts for the recovered instance.

SQL Server in a stand-alone environment

This section provides the BMR backup and recovery procedures for SQL Server in a stand-alone environment.

Backing up a SQL Server for BMR

Configure a SQL Server client resource to back up SQL Server for BMR.

Creating client resources by using the NetWorker Client Backup Configuration Wizard

Create separate client resources to back up the application data and the file system data. The client resource for the file system data must include the `ALL` save set.

Access privileges for backup and recovery on page 44 provides information about how to create a client resource to back up the application data.

The "Creating a client resource with the Client Backup Configuration Wizard" section in the *NetWorker Administration Guide* provides information about how to create a client resource to back up the file system data.

Creating client resources by using the Client Properties dialog box

To back up the application data and the file system data, create separate client resources. The client resource for the file system data must include the `ALL` save set.

Procedure

1. In the Administration view of the NetWorker Management Console, create a policy, a workflow, a group, and an action for Active Directory.

   The Data Protection Policies chapter in the *NetWorker Administration Guide* provides information.

2. To back up the `ALL` save set, create a client resource:

   a. In the **NetWorker Administrator** window, click **Protection**.

   b. In the expanded left panel, right-click **Clients** and select **New**.

      The **Create Client** dialog box appears.

   c. In the **Name** field, type the name of the client resource.

   d. Click **OK**.

3. In the right panel, right-click the client resource and select **Modify Client Properties**.

4. In the **Client Properties** dialog box:

   a. On the **General** tab:

   - In the **Group** list, select the group that you have created in step 1.

   - In the **Save set** field, type **ALL**.

   b. On the **Apps & Modules** tab, ensure that the **Backup command** and **Application information** fields are clear.

   c. Specify other fields according to the requirements.

   d. Click **OK**.

5. Perform the backup.

   Ensure that the backup successfully completes.

6. To back up the `MSSQL:` save set, create a client resource:

   Perform step a to step d of step 2.

7. In the right panel, right-click the client resource and select **Modify Client Properties**.

8. In the **Client Properties** dialog box:

   a. On the **General** tab:

   - In the **Group** list, select the group that you have created in step 1.

   - In the **Save set** field, type the name of the SQL Server instance that contains the databases you must restore:

     - `MSSQL:` for the default SQL Server instance.

     - `MSSQL$<named_instance>` for a named SQL Server instance.

    b. On the **Apps & Modules** tab, in the **Backup command** field, type `nsrsqlsv`.

    c. Specify other fields according to the requirements.

    d. Click **OK**.

9. Perform the backup.

    Ensure that the backup successfully completes.

Performing BMR of a stand-alone SQL Server

The BMR process restores only the `C:` drive. It does not restore the SQL data or the system databases, such as master, msdb, and model.

Procedure

1. Perform the procedures that the "Performing a Windows BMR to physical or virtual computers" section in the *NetWorker Administration Guide* describes.

2. Rebuild the SQL Server system databases.

   Refer to Microsoft documentation for information about how to rebuild the SQL Server system databases. The following table provides links to relevant Microsoft documentation for various versions of SQL Server.

   Table 41 Rebuilding SQL Server system databases

SQL Server version	Microsoft article
SQL Server 2017 and 2016	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.130).aspx
SQL Server 2014	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.120).aspx
SQL Server 2012	https://msdn.microsoft.com/en-us/library/dd207003(v=sql.110).aspx

3. Restore the databases by using the NMM Microsoft SQL Server Management Studio (SSMS) plug-in GUI:

   a. Open the NMM SSMS plug-in GUI.

   b. On the **Database Restore** tab, under **Select a page**, click **General**.

   c. Under **Sources**, make the following selections:

   - Select the NetWorker server.
   - Select the SQL Server host.
   - Select the SQL Server instance.
   - In the list of backed-up records that appears, select the databases to restore.

   d. From the NMM SSMS plug-in GUI, start the restore of the databases.

CHAPTER 11

Troubleshooting

This chapter includes the following sections:

Error logs for backup and recovery

To help you diagnose problems, certain information is written to an application-specific log file during backup and restore operations.

The following types of information are available:

- Software configuration information

- Operation parameters

- Operation status and error messages

Application logs are generated for the `nsrsqlsv.exe` and `nsrsqlrc.exe` programs, and are created in the `nsr\applogs` folder on the SQL Server host. NMM logs are cumulative and are appended each time the `nsrsqlsv.exe` or `nsrsqlrc.exe` programs run. Debug logs are generated each time the `nsrsqlsv.exe` or `nsrsqlrc.exe` programs are run with debug logging enabled. A log is truncated when disk space is exhausted. The following tables cross-reference the program and log file names.

Table 42 Program and log file names without debug logging enabled

Program	Log file
nsrsqlsv.exe	nsrsqlsv.log
nsrsqlrc.exe	nsrsqlrc.log

Table 43 Program and log file names with debug logging enabled

| Program	Log file
nsrsqlsv.exe	nsrsqlsv_*\<timestamp\>*.*\<processid\>*.log
nsrsqlrc.exe	nsrsqlrc_*\<timestamp\>*.*\<processid\>*.log

The logging capabilities of `nsrsqlsv` and `nsrsqlrc` are not cluster-aware. For both clustered and nonclustered configurations, the logs are stored on a local disk.

SQL savegroup notifications

A notification provides information about events that occur in the NetWorker and NMM environments. By default, the NetWorker server sends notifications to log files that are located in the `install_dir\applogs` directory.

NMM supports SQL savegroup notifications that list the backup success or failure and provides important details about any failure.

The following figure shows an example of SQL savegroup notification.

Figure 36 Example of SQL savegroup notification
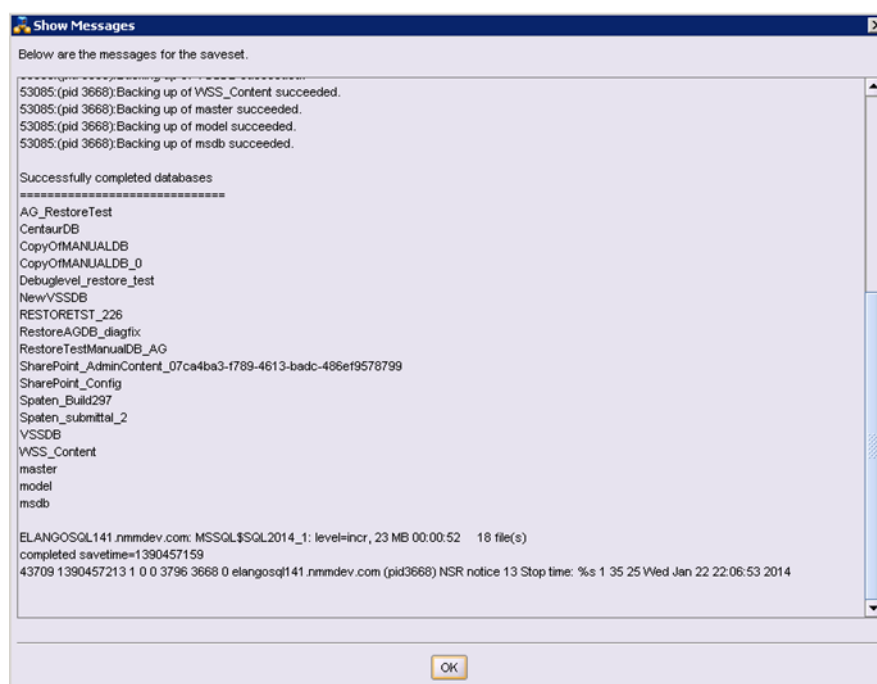
Troubleshooting general issues

"An internal error occurred"

If this error message appears in the operational log, ensure that the user starting the backup or restore operation has the "Read & execute" and "List folder contents" permissions to the database data and log folder. The default data and log folder path is `C:\Program Files \Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\`

"bsVdiAbort: SignalAbort failed: protocol error"

If this error message appears in the operational log, the user is not assigned the "Create global objects" Windows user policy. This policy is required to create named file mapping objects. Access privileges for backup and recovery on page 44 provides information on how to assign Windows user privileges.

Troubleshooting GLR

View the log files to troubleshoot GLR, which are by default located at `C:\Program Files\EMC NetWorker\nsr\applogs`. Set a higher debug level in the **Options** view of the NMM SSMS plug-in GUI for more detailed logs for troubleshooting.

This section contains solutions to common problems.

Mount failure because save set is not GLR-capable

While mounting a backup for GLR by using the SSMS plug-in, the **Monitor** page provides messages about the status of the mount operation. If the backup selected for table-level restore is not a GLR capable backup, then the operation fails and the following error message appears:

```
Error: Saveset '<saveset_ID>' is not GLR capable.
```

SQL GLR failing to establish a Client Direct session

Client Direct is required for NMM SQL GLR. Ensure that Client Direct is enabled before continuing with GLR.

To verify that a Client Direct session is established, use the following steps:

1. Validate that the NetWorker device is enabled for Client Direct.
   This verification must only be performed for AFTD devices. Data Domain is automatically enabled for Client Direct. The *NetWorker Administration Guide* provides more information about Client Direct.

2. Validate that the client has name resolutions for the systems.
   If Data Domain is being used, ensure that the client has name resolution for the Data Domain device. If an AFTD storage node is being used, ensure that the client has name resolution for the storage node.

3. Check the application logs directory in the NetWorker Virtual File System (NWFS) log file, `nwfs.raw`, and look for messages confirming that a Client Direct session was established.

   - The message "`Successfully established a client direct recover session for saveset`" confirms that a Client Direct session is successfully established.

   - The message "`Could not establish a client direct recover session for saveset '<saveset_ID>', exiting`" indicates that a Client Direct session could not be established.

Cannot enable a Client Direct session and GLR failing as a result

Client Direct is required for GLR. However, if you cannot enable Client Direct for either policy or technical reasons, use the following workaround to allow GLR to continue without Client Direct.

1. On the NMM Client host, create a folder and name it "debug" in the `\nsr\` directory, if the folder does not already exist.

2. Within the "debug" folder, create an empty file and name it "nodirectfile" with no file name extension.
   You may be required to create the "nodirectfile" file from a DOS Shell command line.

(i) Note: This workaround disables Client Direct for all client operations, including subsequent backups. This workaround is against NMM best practices and you may run into timeout and other restore issues if you do not enable Client Direct.

"Saveset '*<saveset_ID>*' is not GLR capable"

This error message is logged in the `nwfs.raw` file when the save set you are trying to perform GLR on is not GLR capable. The Considerations on page 128 section provides information on supported backup types.

Table Restore tab does not appear in the NMM SSMS plug-in GUI

If the **Table Restore** tab is not visible in the **NetWorker** window of the NMM SSMS plug-in GUI, verify that ItemPoint is installed in Programs and Features. To install ItemPoint, the SQL GLR option must be selected while installing or changing installation of NMM with the Install Wizard.

Large databases cause slow loading times

ItemPoint may appear to stop responding while reading a backup of a very large database. The time for ItemPoint to read this data from the backups may take hours because of suboptimal Virtual File System performance.

If this issue occurs, perform a SQL flat file restore of the same backup to a server with enough space, and then use ItemPoint on the files on the local disk.

The Considerations on page 128 section provides more information.