Quest

Quest® NetVault® Backup Plug-in *for SQL Server* 12.0

User's Guide

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept.
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our website (https://www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal.

Trademarks

Quest, the Quest logo, Join the Innovation, and NetVault are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit https://www.quest.com/legal/trademark-information.aspx. All other trademarks and registered trademarks are property of their respective owners.

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT NOTE, **NOTE**, **TIP**, **MOBILE**, or **VIDEO:** An information icon indicates supporting information.

NetVault Backup Plug-in *for SQL Server* User's Guide
Updated - June 2018
Software Version - 12.0
MSG-101-12.0-EN-01

Contents

1

Introducing NetVault Backup Plug-in *for SQL Server*

NetVault Backup Plug-in *for SQL Server*: at a glance

Quest® NetVault® Backup Plug-in *for SQL Server* (Plug-in *for SQL Server*) increases confidence in recoverability of SQL Server databases and eliminates the need for complex scripting. Through a web-based user interface (WebUI) and automated workflow process, the plug-in offers a centralized way to set up, configure, and define backup and restore policies. Support for **Online Virtual Device Interface (VDI)** and **Volume Shadow Copy Service (VSS)** backups lets you implement your preferred backup method, without requiring you to learn SQL Server internals. Plug-in *for SQL Server* offers a detailed level of control, which lets you restore complete databases, partial databases, or individual data files, thus minimizing downtime. In addition, the plug-in supports SQL Server features such as Failover Clustering (Active/Passive), AlwaysOn Availability Groups (SQL Server 2012 and later), and Transparent Data Encryption (TDE).

Through integration with a range of backup devices, your data is protected and stored offsite to meet your disaster-recovery and business-continuity goals.

Key benefits

- **Increase confidence when deploying the plug-in:** With Plug-in *for SQL Server*, you do not have to worry about learning SQL Server internals before implementing a backup policy that accounts for many recovery scenarios. You can choose the best method for the job. You might choose VDI, which provides maximum reliability and performance while supporting the full range of SQL Server backup-and-restore functionality. You might choose Microsoft's VSS framework, which lets you perform volume backups while applications on a system continue to write to the volumes, all without writing complex scripts.

  Plug-in *for SQL Server*'s backup features also include:

  - Protection for standalone and cluster environments
  - Online VDI- or VSS-based backups
  - Full and Differential Database backups while data is online and accessible
  - Incremental Transaction Log and Tail-Log backups (VDI only)

- Copy-Only backups
- Advanced Full and Differential File backups along with Partial Database and Differential Partial Database backups
- Protection for FILESTREAMs
- Protection down to the datafile level
- Backup-compression support
- Creation of backup checksums that are used during restores to detect corruption

By relying on the plug-in to implement your backup policies, you can focus on more critical tasks without risking your ability to recover what is needed if a failure occurs. In addition, the IT manager's confidence is increased by knowing that SQL Server data is protected.

- **Speed up restores to reduce downtime:** Plug-in *for SQL Server* empowers you to create a comprehensive and flexible backup policy and reduces the need for scripting, which may increase the risk of syntax or human errors. Select what needs to be restored, the backup set to restore from, and, if appropriate, the time or marked transaction restore point, and the plug-in performs recovery without further interaction.

  Additional Plug-in *for SQL Server* restore-and-recovery features include:

  - Full, differential, incremental, and time and marked transaction point-in-time (PIT) restores
  - Restores of complete databases, partial databases, or individual datafiles
  - Rename of datafiles
  - Restore VDI-based backups to an alternate server
  - Restore VDI-based backups from a clustered configuration to a standalone installation
  - Disaster recovery
- **Ensure business continuity:** With offsite backups being an important part of the data-protection for business-critical applications, the plug-in takes advantage of NetVault Backup's integration with a range of backup devices. NetVault Backup lets you select which backup device to store the backup on. You can store the backup online in a virtual tape library (VTL). You can also duplicate the job to physical tape libraries shared by multiple SQL Server databases, other proprietary databases, or used for backup.

  Plug-in *for SQL Server* gives you the confidence that your SQL Server environment is protected and stored offsite for disaster-recovery purposes. At the same time, it frees administrators from being present 24x7 as less-experienced personnel can initiate restores.

Feature summary

- Protection for standalone and cluster environments
- Online VDI- or VSS-based backups
- Full and Differential Database backups while data is online and accessible
- Incremental Transaction Log and Tail-Log backups (VDI only)
- Copy-Only backups
- Advanced Full and Differential File backups along with Partial Database and Differential Partial Database backups
- Protection for FILESTREAMs
- Protection down to the datafile level
- Backup-compression support
- Creation of backup checksums that are used during restores to detect corruption

- Full, differential, incremental, and time and marked transaction PIT
- Verify-only restore option
- Restores of complete databases, partial databases, or individual datafiles
- Rename of datafiles
- Restore VDI-based backups to alternate server
- Disaster recovery
- Point-and-click WebUI

Target audience

This guide is intended for users who are responsible for the backup and recovery of SQL Server. Familiarity with SQL Server administration is assumed. Advanced knowledge of SQL Server is useful for defining an efficient backup-and-recovery strategy and performing advanced recovery scenarios.

Recommended additional reading

Quest recommends that you have the following documentation available for reference while setting up and using this plug-in.

- **SQL Server documentation:**
  - SQL Server 2017 Books Online: https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017
  - SQL Server 2016 Books Online: https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=sql-server-2016
  - SQL Server 2014 Books Online: http://technet.microsoft.com/en-us/library/ms130214.aspx
  - SQL Server 2012 Books Online: http://technet.microsoft.com/en-us/library/ms130214(v=sql.110).aspx
  - SQL Server 2008 R2 Books Online: http://technet.microsoft.com/en-us/library/ms130214(v=sql.105).aspx
  - SQL Server 2008 Books Online: http://technet.microsoft.com/en-us/library/ms130214(v=sql.100).aspx
  - Volume Shadow Copy Service: http://msdn.microsoft.com/en-us/library/bb968832.aspx
- **NetVault Backup documentation:**
  - *Quest NetVault Backup Installation Guide*: This guide provides details on installing the NetVault Backup Server and Client software.
  - *Quest NetVault Backup Administrator's Guide*: This guide explains how to use NetVault Backup and describes the functionality common to all plug-ins.
  - *Quest NetVault Backup CLI Reference Guide*: This guide provides a description of the command-line utilities.

  You can download these guides from https://support.quest.com/technical-documents.

2

Planning your SQL Server deployment

Deployment overview

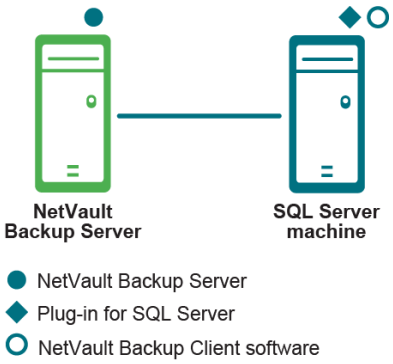
Microsoft supports the deployment of SQL Server on a single server or in a high-availability environment. With high-availability environments, Microsoft supports Failover Clustering in an Active/Passive configuration (SQL Server 2014 and earlier), which is based on the Microsoft Cluster Service (MSCS). Microsoft also supports Failover Clustering in an Active/Active configuration with AlwaysOn Availability Groups (SQL Server 2012 and later), which is based on Windows Server Failover Clusters (WSFC).

Deploying the plug-in in these environments is almost identical because the Plug-in *for SQL Server* is installed on the server that hosts the SQL Server database. The following topics describe how the Plug-in *for SQL Server* is deployed in each type of SQL Server deployment.

Standalone deployment

You can set up one machine as both the NetVault Backup Server and the SQL Server, that is, all software installation and configuration requirements are performed on a single machine. However, Quest recommends that these two entities exist on separate machines.

Figure 1. Standalone deployment



Important considerations for standalone deployment

- Regardless of the environment in place, two entities or one, install Plug-in *for SQL Server* on the host where SQL Server resides.
- If you use SQL Server Replicated Databases, obtain the required number of plug-in licenses based on the replication type and the backup and recovery requirements. Also install the plug-in on each host included in the SQL Server Replication environment, including the following:
  - **Publisher:** Install the plug-in on this node regardless of the replication type implemented.
  - **Distributor:** If you use a Local Distributor, the Distributor is running on the same node as the Publisher. However, if you use a Remote Distributor, install the plug-in on the Distributor host.
  - **Subscriber:** If you want to back up the changed data in a Subscriber database, install the plug-in on the **Subscriber** hosts. This configuration lets you synchronize the Publication database with the Subscription database after recovery. If the plug-in is not installed on the Subscriber nodes, re-initialize all subscriptions to the publications in the Publication database after recovery.

  The plug-in considers each of these nodes as an individual Client.

High-availability deployments

Whether you choose to deploy an Active/Passive or Active/Active configuration, Microsoft requires that you install and configure failover clustering. High-availability deployments include:

- Failover Cluster (Active/Passive) with SQL Server 2008 R2 and later
- AlwaysOn Availability Group (Active/Active) with SQL Server 2012 and later

Important considerations for high-availability deployments

- This guide does not offer instructions on how to set up NetVault Backup's **Application Cluster Support** to administer backups and restores of non-SQL Server-related data and files. This process is not plug-in-specific, and you can find complete details in the *Quest NetVault Backup Administrator's Guide*.
- Before you continue, Quest recommends that you review all cluster-related information provided in the *Quest NetVault Backup Administrator's Guide*. That guides helps you understand how the information included in this guide works with SQL Server Failover Cluster and AlwaysOn Availability Group functionality.
- When interacting with an AlwaysOn Availability Group, Quest recommends that you use the Fully Qualified Domain Name (FQDN) of the cluster. You can also use the listener IP or IP address of the cluster.
- When configuring AlwaysOn Availability Group or Failover Clustering of SQL Server, in the configuration of the Virtual Client, use the configured listener IP of the SQL Server instead of IP address of the cluster.

Failover Clustering

SQL Server Failover Clustering (Active/Passive) provides high-availability for an entire SQL Server Instance. For example, you can configure a SQL Server Instance on one node of a failover cluster to fail over to a different node in the cluster during a failure or planned upgrade.

A failover cluster is a combination of one or more nodes (hosts) with two or more shared disks, known as a **resource group**. The combination of a resource group, its network name, and an IP address that makes up the clustered application or server is called a **Virtual Server**. A Virtual Server appears on the network as if it were a single computer, but provides failover from one node to different node if the current node becomes unavailable.

IMPORTANT: In NetVault Backup terminology, a cluster node is called a **Virtual Client**. The references to **Virtual Client** in Plug-in *for SQL Server* are basically references to the **Virtual Server** in a SQL Server Failover Cluster environment or AlwaysOn Availability Group.

Using the failover cluster network name, the Plug-in *for SQL Server* identifies the current node that is in control of the SQL Server Virtual Server and targets it for backup.

Make sure that you use the same Windows Server to host the Cluster Core Resources group and to take the active role. The Cluster Core Resources group contains the IP address, network name, and the disk witness. For the Virtual Client to function correctly, the Windows Server that hosts the Cluster Core Resources group, that is, the host identified as the Current Host Server, *must* be the same node that holds the active role. If failover occurs and the active role moves to a different host but the Cluster Core Resources group does not, the Virtual Client cannot access the active host. The Virtual Client must resolve the IP address for the cluster to the server that takes the active role.

If necessary, such as after a failover occurs, use Windows PowerShell or a command prompt to move the Cluster Core Resources group to the active host.

PowerShell example: `Move-ClusterGroup "Cluster Group" –node <ClusterNodeName>`

Command prompt example: `cluster group "Cluster Group" /Move:<ClusterNodeName>`

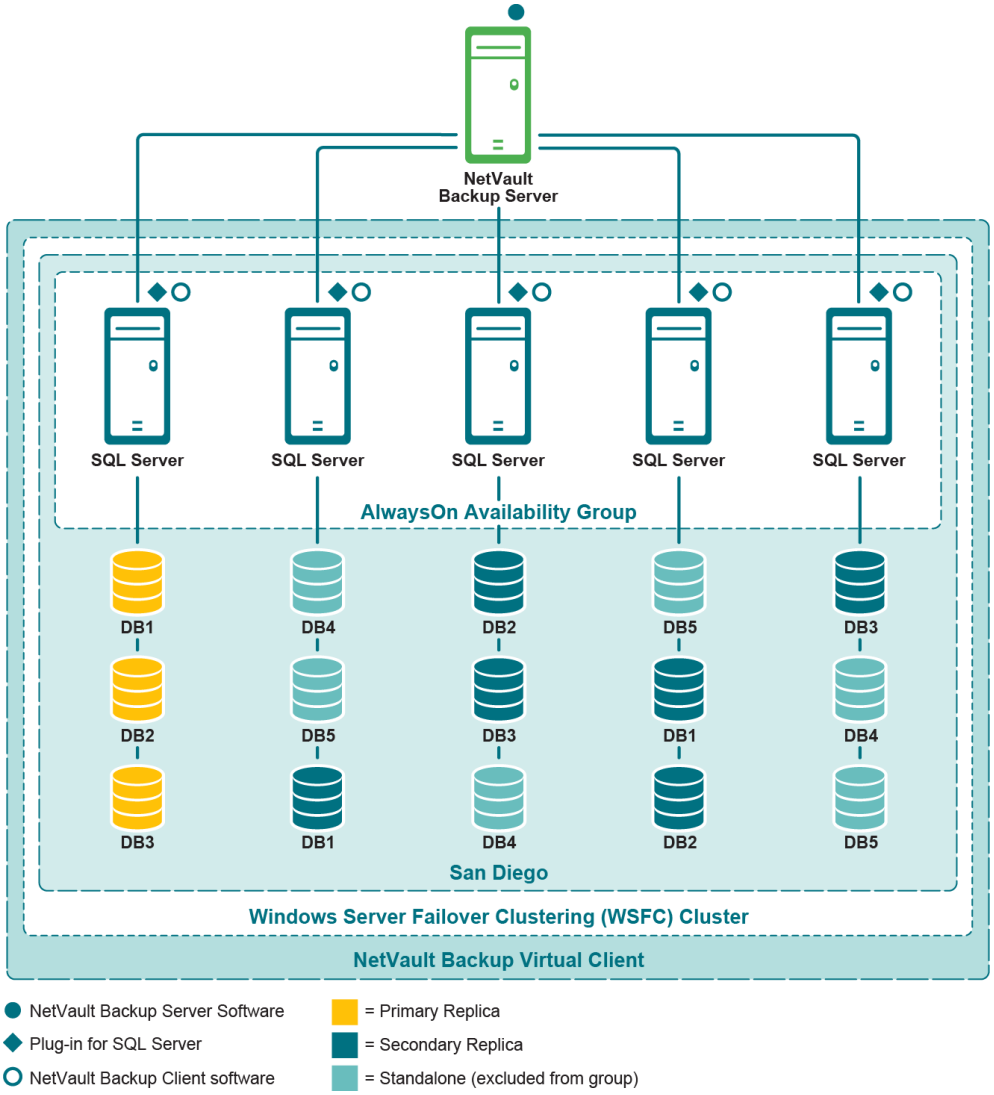
AlwaysOn Availability Group

You can use the plug-in with the AlwaysOn Availability Groups that you have created on top of your WSFC cluster. In addition to backing up data, you can use the plug-in to manage the addition and removal of the primary and secondary replicas during a restore process. This option eliminates the need to use SQL Server Management Studio to add and remove the replicas.

To ensure that the plug identifies a Virtual Client as running on an AlwaysOn Availability Group, enter valid credentials for the All Instances node located under the applicable Virtual Client in the selection tree. The credentials must let the plug-in log in to at least one SQL Server Instance that is a member of the group. For more information, see Configuring the plug-in.

NOTE: Quest recommends creating backups that include no more than 100 databases in an AlwaysOn Availability Group. There is not an enforced limit for the maximum number of Availability Groups and Availability Databases per machine. The actual number of Databases depends on the hardware capabilities, resources, and workload. However, Microsoft documents that extensive tests have been conducted with 10 Availability Groups and 100 Databases per physical machine. For more information, see: https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/prereqs-restrictions-recommendations-always-on-availability

Figure 2. AlwaysOn deployment



AlwaysOn Failover Cluster Instances

You can use the plug-in with a SQL Server AlwaysOn Availability Group cluster consisting of two nodes, with each node itself being a WSFC failover cluster, with the two failover clusters at different physical locations, and with only manual failover allowed.

In this case, one of the instances, for example “SQLinstance” is an instance of SQL Server running on the failover cluster with the “primary” role of the AlwaysOn Group. The other instance, for example “SQLDRinstance” is an instance of SQL Server that is running on the failover cluster with the “secondary” role.

If the logins are the same for each instance, then enter valid credentials for the All Instances node located under the applicable Virtual Client in the selection tree.

If the logins are different for each instance, then enter valid credentials for respective instance, that is listed under the All Instances nodes. If the instances are not listed, use the **Add Instance** action, to enter the credentials for each instance.

When creating the NetVault Backup Virtual Client for AlwaysOn Failover Cluster Instances, provide the IP address of the Virtual Network Name (the “Virtual IP”) of one of the two failover cluster instances that consists the AlwaysOn group. The Virtual IP address of the instance with either the primary role, or the secondary role may be provided. However, if the instance with secondary role is at a remote location from the NetVault Backup Server, Quest recommends using the Virtual IP address of the instance with the primary role for improved performance.

When running backups, you must set the replica selection algorithm as Primary, which is the default algorithm. Backups running using the Secondary replica selection algorithm might perform slow, due to the secondary instance being at a remote location from the NetVault Backup Server.

NetVault Backup Policy Management

A NetVault Backup Job Policy can be used to submit one or more jobs that target one or more similar clients.

If you intend to use NetVault Backup Job Policy across your organization's SQL Server clients, Quest strongly recommends that you use, whenever possible, a generic name for the SQL Server Instances deployed in different SQL Servers. Do not use a name associated with the machine on which the instance resides, or a name that is unique.

Jobs generated for instances with unique names, in general, cannot be ported to other SQL Server clients in your organization. Using generic instance names improves portability and policy management across all affected clients. In this way, you can create NetVault Backup Job Policies that can be applied to the different SQL Server clients in your organization.

For more information, see the *Managing Policies* in the *Quest NetVault Backup Administrator's Guide*.

3

Installing and removing the plug-in

Installation prerequisites

The requirements vary depending on your configuration:

Prerequisites for a standalone deployment

Before installing Plug-in *for SQL Server*, check that the following software is installed and configured on the machine that is to host SQL Server:

- **NetVault Backup Server and Client software:** At a minimum, the Client version of NetVault Backup software must be installed on the machine configured as the SQL Server.
- **SQL Server software:** The machine must be running a supported version of SQL Server. For a list of supported versions, see the *Quest NetVault Backup Compatibility Guide*, which is available at https://support.quest.com/technical-documents.
- **SQL Server VSS Writer Services:** If you want to implement a VSS backup strategy for SQL Server databases, the **SQL Server VSS Writer** service must be running. The **SQL Server VSS Writer** is automatically installed. However, the service is not set to start automatically. To enable VSS backups, use the Windows Services utility. Quest recommends that you set the Startup Type for this service to **Automatic**.

Additional prerequisites for a high-availability deployment

Before installing Plug-in *for SQL Server*, check that the requirements are met:

- **Microsoft SQL Server Failover Clustering or AlwaysOn environment in place:** You must have a properly configured environment.

- Failover Clustering is only supported with SQL Server Standard and Enterprise Editions. For more information on installing Failover Clustering, see the *Failover Clustering* section of *SQL Server 20xx Books Online*.
- AlwaysOn requires SQL Server 2012 and later and is supported on Windows 2008 R2 and later. For more information, see the applicable Microsoft documentation: http://msdn.microsoft.com/en-us/library/hh510230(v=sql.120).aspx
    - For SQL Server 2012, see: http://msdn.microsoft.com/en-us/library/hh510230(v=sql.110).aspx
    - For SQL Server 2014, see: http://msdn.microsoft.com/en-us/library/hh510230(v=sql.120).aspx

- **Separate NetVault Backup Server machine:** The machine that serves as the NetVault Backup Server must be configured and it ***must exist outside*** the SQL Server cluster. The machine must have network connectivity to the nodes (hosts) within the cluster.

Installing or upgrading the plug-in in a standalone environment

You can install or upgrade the plug-in on individual systems, one at a time. You can also deploy the plug-in to multiple machines by creating a deployment task from the WebUI to push the packages to the specified machines. You can use this method for both new and upgrade installations on Windows-based machines. After the packages are installed on the machines, the new machines are automatically added to the NetVault Backup Server.

- Performing a local installation
- Performing a remote installation

Performing a local installation

1. In the Navigation pane, click **Manage Clients**.
2. On the **Manage Clients** page, select the machine that contains the SQL Server, and click **Manage**.
3. On the **View Client** page, click the **Install Plugin** button ( ).
4. Click **Choose Plug-in File**, navigate to the location of the **".npk"** installation file for the plug-in, for example, on the installation CD or the directory to which the file was downloaded from the website.

   Based on the OS in use, the path for this software may vary on the installation CD.
5. Select the file entitled **"sql-x-x-x-x-x.npk,"** where **xxxxx** represents the version number and platform, and click **Open**.
6. To begin installation, click **Install Plugin**.

   After the plug-in is successfully installed, a message is displayed.

Performing a remote installation

This process is not plug-in-specific. For more information, see the topic on deploying clients and plug-in packages in the *Quest NetVault Backup Administrator's Guide*.

1. In the Navigation pane, click **Guided Configuration**.
2. On the **NetVault Configuration Wizard** page, click **Install Software/Add Clients**.
3. On the **Select Software/Add Clients** page, select **Install NetVault software on remote machines**.

4 In the **Package Store** list, select the applicable repository.

5 Click **Add NetVault plug-in package**.

6 In the **Select packages for deployment** dialog, select the file entitled **"sql-x-x-x-x-x.npk,"** where **xxxxx** represents the version number and platform, and click **OK**.

The plug-in file is listed in the deployment table.

7 Click **Next**.

8 On the **Machines to Have NetVault Software Installed** page, select the applicable machines.

9 If you are installing an upgrade, select **Allow existing NetVault client installations to be upgraded**.

10 Click **Install Software/Add Clients**.

The status is displayed on the next page.

11 After the plug-in is successfully installed, click **Next** to complete the process.

Installing or upgrading the plug-in in a high-availability environment

Installation of the plug-in in a high-availability environment is completed through the creation of a **Virtual Client** on the NetVault Backup Server. A Virtual Client is a group of nodes within the cluster. NetVault Backup Server views this group as a ***single*** client that is created to back up a single clustered resource, for example, a SQL Server Virtual Server. During the Virtual Client creation process, the plug-in is transferred from the NetVault Backup Server to selected nodes within a cluster and installed there.

Creating a Virtual Client

As noted earlier, the Virtual Client creation process is not plug-in-specific, and you can find complete details in the *Quest NetVault Backup Administrator's Guide*. However, consider the following points during the Virtual Client creation process:

- **Assign a name to the Virtual Client:** Quest strongly recommends that you use the Virtual Server network name (that is, the Fully Qualified Domain Name [FQDN]) assigned to the SQL Server as the NetVault Backup Virtual Client name. When you browse a Virtual Client, NetVault Backup locates the node currently in control of the clustered application, and reveals the SQL Server Instance, for example, on the **NetVault Backup Selections** page. With a Virtual Client name set up as the SQL Server Virtual Server network name, you can recognize the SQL Server Instance for which the Virtual Client was created.
- **Only include relevant cluster nodes in the Virtual Client:** Include only the hosts that are relevant to the SQL Server Virtual Server that is to be backed up and restored.

After the creation of the Virtual Client, the plug-in is transferred to all designated cluster nodes and installed locally. You use the installed plug-in by using the Virtual Client to back up and restore shared data; you can ***only*** perform backups and restores of data shared within the cluster.

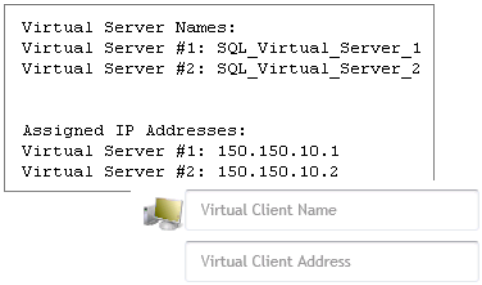
Using multiple SQL Server Virtual Servers in the same cluster

SQL Server supports the ability to create multiple Virtual Servers in a cluster. However, each Virtual Server can have only one instance of SQL Server running. In addition to the provisions outlined earlier, account for the following provisions when using Plug-in *for SQL Server* in this configuration.

Create a NetVault Backup Virtual Client for each SQL Server Virtual Server. Each Virtual Server has its own network name and IP address, and it is displayed as a separate Virtual Client in NetVault Backup.

- **Create an individual NetVault Backup Virtual Client for each SQL Server Virtual Server:** For each SQL Server Virtual Server in the cluster, create a separate Virtual Client. During the creation process, ensure that the following are included:
  - **IP Address of the SQL Server Virtual Server:** In the **Virtual Client Address** field of the **Virtual Client** page, enter the IP Address assigned to the appropriate SQL Server Virtual Server. For example, if you are creating the first of the two SQL Server Virtual Servers in a failover cluster, enter the IP address assigned to the ***first*** Virtual Server.
  - **Give a name to the Virtual Client that corresponds to the network name of the SQL Server Virtual Server:** Enter the network name associated with the SQL Server Virtual Server in the **Virtual Client Name** field.

  Figure 3. Creation of Virtual Client for first SQL Server Virtual Server in clustered environment that contains Virtual Servers

- **Copy the "nvsqlserver.cfg" file before creating the next NetVault Backup Virtual Client:** Because creating a new Virtual Client overwrites the configuration file, which includes the Logon Detail information, Quest recommends that you perform the following steps:
  1. Store a copy of the first **"nvsqlserver.cfg"** file in a safe location.

     You can find this file in: \\<*NetVaultBackupInstallDirectory*>\config
  2. Create the next Virtual Client, store a copy of its **"nvsqlserver.cfg"** file in a safe location, and repeat this step for every new Virtual Client that you need to create.

     Ensure that the correct network name for the SQL Server Virtual Server and IP address is provided for each.
  3. When you have finished creating Virtual Clients, copy the information from the **[Security-<*instanceName*>]** and **[ServerList:List]** sections in each **"nvsqlserver.cfg"** file to the same sections in the **"nvsqlserver.cfg"** file on the primary node.
  4. If you have browsed the Virtual Client, close and reopen the **NetVault Backup Selections** page.

Licensing the plug-in

The plug-in supports backup and restore of shared data ***only***. For a SQL Server Failover Cluster or AlwaysOn environment, only a single license is needed—a clustered application license for the Virtual Client.

For information on the licensing process, including how to obtain the proper license keys, see the *Quest NetVault Backup Installation Guide*.

Removing the plug-in from a standalone deployment

For details on removing the Plug-in *for SQL Server* in high-availability deployment, see the applicable topic on working with client clusters in the *Quest NetVault Backup Administrator's Guide*.

> **CAUTION: Removing the plug-in does not remove the configuration settings that you created. If you reinstall the plug-in or install a newer version, the plug-in typically uses the same configuration settings from the previous installation unless otherwise indicated in the release notes.**

1. In the Navigation pane, click **Manage Clients**.
2. On the **Manage Clients** page, select the applicable client, and click **Manage**.
3. In the **Installed Software** table on the **View Client** page, select **Plug-in *for SQL Server***, and click the **Remove Plugin** button ( ).
4. In the **Confirm** dialog box, click **Remove**.

Removing a SQL Server Instance

1. In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.
2. In the selection tree, open the applicable client node.
3. Double-click **Plug-in *for SQL Server***, double-click the **All Instances** node, and click the applicable instance to select it.
4. From the context menu, select **Remove**.
5. When the confirmation message appears, click **Yes**.

4

Configuring the plug-in

Configuring authentication details

Before initiating a backup, configure the plug-in's authentication details, which include authentication mode, user name, and password. Use the Configure dialog to specify this information.

How the information that you enter in the Configure dialog applies depends on whether you are configuring a client for a standalone deployment or a NetVault Backup Virtual Client for a SQL Server Failover Cluster or an AlwaysOn Availability Group. When you enter settings for a standalone client, the settings apply to all backups generated from that client when it is selected in the selection tree. If you use Virtual Clients to support a SQL Server Failover Cluster or an AlwaysOn Availability Group, the authentication information that you enter for a specific instance in the cluster or group is used for all nodes within the same cluster or group. All information that you enter on the Configure dialog for a cluster or group is maintained on the NetVault Backup Server, unlike a standalone deployment for which the information is maintained on the server on which the plug-in is installed.

Authentication modes

SQL Server provides two authentication modes: Windows Authentication and SQL Server Authentication.

- **Windows Authentication:** With Windows Authentication, you connect through a Microsoft Windows user account. SQL Server validates the account name and password using information in the Windows OS. This method is the default, and is much more secure than ***Mixed Mode***, which allows a combination of Windows authentication and SQL Server Authentication.

  Windows Authentication uses Kerberos security protocol, provides password policy enforcement in terms of complexity validation for strong passwords, supports account lockout, and supports password expiration. Microsoft strongly recommends implementing a Windows Authentication mode for SQL Server.

- **SQL Server Authentication:** With SQL Server Authentication on SQL Server 2008 or earlier, you must provide the user name and password for a SQL Server user with the **sysadmin** role. With SQL Server Authentication on SQL Server 2012 or later the **sysadmin** role is not supported. However, you can assign the **sysadmin** role to a domain user account, such as Administrator, or you can set the SQL Server service to run using the **Log on as** option set to a domain user that has the privileges. For a system that is not part of a domain, you can assign the role to a local user or you can set the SQL Server service to run under the

local user that has the privileges. For more information, see Setting up SQL Server Authentication for SQL Server 2012 or later.

Configuring the NetVault Backup Virtual Client

After you install (or reinstall) a NetVault Backup Virtual Client and before you perform a backup or restore, ensure that the NetVault Backup Virtual Client is configured. The process for configuring the Virtual Client depends on whether you are using a SQL Server Failover Cluster or an AlwaysOn Availability Group.

Configuring the Virtual Client for a SQL Server Failover cluster deployment

In a SQL Server Failover deployment, the SQL Server Failover Virtual Instance is automatically detected if the SQL Server Browser Service is running in all the nodes of the Failover Cluster. Otherwise, to add SQL Server Failover Virtual Instance using the **Add Instance** Action, follow these steps:

1. In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.
2. In the selection tree, open the newly created Virtual Client.
3. Double-click **Plug-in *for SQL Server***, and double-click the **All Instances** node.
4. From the context menu, select **Configure**.
5. Complete the applicable fields. For more information, see Configuring authentication details.

   IMPORTANT: Add the cluster instance name in the **Instances** field of the **Configure** dialog box. To add an instance, if necessary, you might specify the SQL Server cluster name as VIRTUAL SERVER NAME\INSTANCE NAME

6. To save the settings, click **OK**.
7. Optionally, if you anticipate having to create more backup jobs or modify existing backup jobs on the secondary nodes, perform the following steps:
   a. Fail over the primary node to the secondary node.
   b. Repeat Step 1 through Step 6.
   c. Fail back to the primary node.

If your environment uses multiple Virtual Servers in the same failover cluster, create a NetVault Backup Virtual Client for each Virtual SQL Server, and complete the previous steps for each Virtual Client you created.

Configuring the Virtual Client for an AlwaysOn Availability Group deployment

1. In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.
2. In the selection tree, open the newly created Virtual Client.
3. Double-click **Plug-in *for SQL Server***.
4. In the **Configure** dialog box, enter the credentials of the domain user that is used as the default user for accessing SQL Server Instances in the group.

5 Complete the applicable fields. For more information, see Configuring authentication details.

> **IMPORTANT:** In addition to the initial configuration, you can also enter SQL Server Instance configuration details in the NetVault Backup Virtual Client. You can enter this information at the All Instances level or at the individual instance level. This process works the same way as described in Configuring authentication details.

6 To save the settings, click **OK**.

Configuring plug-in options

The settings described in the following topic apply to all backup and restore jobs and to individual clients. Additional logon settings are also available; for more information, see Entering logon credentials for all instances on a client and Entering default logon credentials for a specific SQL Server Instance.

1 Access the **Configure** dialog box.

- For standalone clients or clients included in an AlwaysOn Availability Group:
  a In the Navigation pane, click **Change Settings**.
  b Click **Server Settings** or **Client Settings**, as applicable.
  c If you selected **Client Settings**, select the applicable client, and click **Next**.
  d On the **Settings** page, click **Plugin Options**.
  e Locate the **Plug-in *for SQL Server*** section of the dialog box.
- For Virtual Clients used in a failover cluster:
  a In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.
  b In the selection tree, open the applicable Virtual Client.
  c Click **Plug-in *for SQL Server***, and select **Configure** from the context menu.

2 Complete the applicable fields:

- **Error Encountered During Checksum:** SQL Server lets you specify the action that should be taken if an error is encountered when performing checksums on backups and restores. You can specify the default action to take on encountering checksum errors by selecting one of the following two options from the list:
  - **Continue after error:** This option is the default set during the plug-in installation. With this value selected, the backup or restore job continues in spite of checksum errors.
  - **Stop on error:** Select this option to set it as the default action. With this value selected, the backup or restore job stops when a checksum error is encountered.
- **New Database Found in Differential/Incremental Backup:** This field applies only to Online VDI backups. If a new database is found while performing **Transaction Log** or **Differential Database** backups and the **Instance Node** is selected on the **NetVault Backup Selections** page, you can ignore it or take a full backup. You can set this action globally for all **Transaction Log** and **Differential Database** backups by selecting the applicable action from the list:
  - **Do full database backup:** This option is the default set during the plug-in installation. Retain this option to perform a full backup of all databases added since the last **Full Database** backup.
  - **Ignore:** Select this option to ***ignore*** databases created since the last **Full Database** backup; that is, databases created after the last backup are left out of the **Transaction Log** or **Differential Database** backup.

- **Undo file for Standby Restores:** For performing Standby Restores, the plug-in requires a temporary Undo File. It is required only when the **Action After Restore** field is set to **With STANDBY**.

  Specify the directory path for the temporary **UNDO.DAT** file in this field.

- **Incomplete Backup of ALL Items Selected:** The plug-in can do one of the following when this error condition occurs:

  - **Complete with Warnings — Saveset Retained:** The job returns a status of **"Backup Completed with warnings"** and a backup saveset is created that includes the items that were successfully backed up.
  - **Complete without Warnings — Saveset Retained:** The job completes and returns a status of **"Backup Completed."** The errors are logged in the NetVault Backup binary logs and ignored on the **Job Status** page. A backup saveset is created that includes the items that were backed up.
  - **Fail — Saveset Retained:** The job returns a status of **"Backup Failed."** However, a backup saveset is generated that includes the items that were successfully backed up.
  - **Fail — No Saveset Retained:** The job returns a status of **"Backup Failed"** and no saveset of backed-up objects is kept. That is, even if some of the objects were successfully backed up, the saveset is discarded.

  If either of the following occurs, the plug-in overrides the **Incomplete Backup of ALL Items Selected** setting and responds as indicated:

  - If a fatal error occurs, the job returns a **"Backup Failed"** status.
  - If a Partial Database or Differential Partial Database backup is performed on a database whose backup target contains a read-only filegroup, the job returns a **"Backup Completed with warnings"** status.

- **Media Format during VSS Restores:** Enter **MTF**, **CPIO**, or **Default**, or leave the field blank.

  - **MTF**, **Default**, or **blank:** If the backup target is tape-based, the plug-in always uses the MTF format. This behavior is used even if a tape target uses Linux or UNIX. MTF is always acceptable for restoring from a backup produced by this version and recent versions of the plug-in. You do not have to enter this setting. Entering **Default** or leaving the field blank is the equivalent of entering MTF.
  - **CPIO:** Do *not* use this setting unless instructed to do so by Quest Support. Quest might ask you to use this setting to restore a backup created and written to a Linux-based tape target with a previous version of the plug-in.

  IMPORTANT: Quest recommends that you leave this option blank and let NetVault Backup determine the format to use. Only change this option if you need to override the default format during a restore from tape that failed. For more information, see Troubleshooting.

- **Check SQL Service Agent Status When Restoring msdb Database**: If the restore job includes the **msdb** database, use this option to instruct the plug-in to check the running status of the SQL Server Service Agent. To restore the **msdb** database and to ensure exclusive access to the database, the SQL Server Service Agent must *not* be running.

  If this option is selected and the restore job includes the **msdb** database, the plug-in checks the status of the SQL Server Service Agent. If the agent is running, the restore job fails before attempting to restore the **msdb** database.

  If this option is not selected, the plug-in does not check the status of the SQL Server Service Agent, and continues the restore job. The job succeeds if the agent is not running and fails if the agent is running.

- **AlwaysOn Availability Groups Cluster**: Use this option to notify the plug-in that your environment consists of AlwaysOn Availability Groups. if AlwaysOn Availability Groups are detected, the plug-in automatically selects this option.

3 To save the settings, click **Apply**.

4 If you are configuring the clients for an AlwaysOn Availability Group, repeat the preceding steps for each client in the group.

Entering logon credentials for all instances on a client

If all instances use the same logon credentials, use this procedure to specify the default credentials. The plug-in also lets you omit the name and password and use the **Log on as** account information defined for the NetVault Process Manager service.

1 In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.

2 In the selection tree, open the applicable client node.

The client node can be a standalone server or Virtual Client.

3 Double-click **Plug-in *for SQL Server***.

4 Click the **All Instances** node, and select **Configure** from the context menu.

5 Complete the applicable fields:

- **Logon security mode:** Based on the ***Authentication Mode*** set for the SQL Server Instance, you can specify either a Windows Administrator or a SQL Server Administrator account. To specify a Windows Administrator account, select **Windows**. For a SQL Server Administrator account, select **SQL Server**.

  IMPORTANT: You can only select **SQL Server** if the ***SQL Server Authentication Mode*** is set to ***Mixed Mode/SQL Authentication***.

- **Login Details:** Depending on the option selected in the **Logon security mode** section, provide the appropriate account information in this section:
  - For Windows authentication, provide the following information:
    - **Administrator User Name:** Specify either a Local or Domain Windows Administrator user name.
    - **Password:** Enter the password associated with the user specified in the preceding field.
    - **Windows Domain:** If a Domain Administrator is specified in the **Administrator User Name** field, enter the name of the domain. You can leave this field blank if a Local Administrator user name is specified.
    - **Login Timeout (Seconds) (0=Timeout disabled):** This option specifies how long to wait, in seconds, before an instance returns from an attempt to log in to a SQL Server Instance. By default, the timeout value is set to 30 seconds; however, you can change the value during a network delay for a particular SQL Server Instance. The maximum that you can use is **800** seconds. If the value is set to **0**, the option is disabled.
  - For SQL Server authentication, provide the following information:
    - **Administrator User Name:** Specify a SQL Server user with the sysadmin role.
    - **Password:** Enter the password associated with the user specified in the preceding field.
    - **Login Timeout (Seconds) (0=Timeout disabled):** This option specifies how long to wait, in seconds, before an instance returns from an attempt to log in to a SQL Server Instance. By default, the timeout value is set to 30 seconds; however, you can change the value during a network delay for a particular SQL Server Instance. The

maximum that you can use is **800** seconds. If the value is set to **0**, the option is disabled.

6 To save the settings, click **OK**.

Configuring an instance at all instances level on a client

The **Add Instance** option in the context menu helps you add an instance under the All Instances node. This option is available if the All Instances node is selected. This option is similar to the Configure option; however, this option lets you specify the exact name of a SQL Server Instance.

1 In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.

2 In the selection tree, open the applicable client node.

The client node can be a standalone server or Virtual Client.

3 Double-click **Plug-in *for SQL Server***.

4 Click the **All Instances** node, and select **Add Instance** from the context menu.

5 Complete the applicable fields:

- **Logon security mode:** Based on the ***Authentication Mode*** set for the SQL Server Instance, you can specify either a Windows Administrator or a SQL Server Administrator account. To specify a Windows Administrator account, select **Windows**. For a SQL Server Administrator account, select **SQL Server**.

  IMPORTANT: You can only select **SQL Server** if the ***SQL Server Authentication Mode*** is set to ***Mixed Mode/SQL Authentication***.

- **Login Details:** Depending on the option selected in the **Logon security mode** section, provide the appropriate account information in this section:
  - For Windows authentication, provide the following information:
    - **Administrator User Name:** Specify either a Local or Domain Windows Administrator user name.
    - **Password:** Enter the password associated with the user specified in the preceding field.
    - **Windows Domain:** If a Domain Administrator is specified in the **Administrator User Name** field, enter the name of the domain. You can leave this field blank if a Local Administrator user name is specified.
    - **Login Timeout (Seconds) (0=Timeout disabled):** This option specifies how long to wait, in seconds, before an instance returns from an attempt to log in to a SQL Server Instance. By default, the timeout value is set to 30 seconds; however, you can change the value during a network delay for a particular SQL Server Instance. The maximum that you can use is **800** seconds. If the value is set to **0**, the option is disabled.
  - For SQL Server authentication, provide the following information:
    - **Administrator User Name:** Specify a SQL Server user with the sysadmin role.
    - **Password:** Enter the password associated with the user specified in the preceding field.
    - **Login Timeout (Seconds) (0=Timeout disabled):** This option specifies how long to wait, in seconds, before an instance returns from an attempt to log in to a SQL Server Instance. By default, the timeout value is set to 30 seconds; however, you can change the value during a network delay for a particular SQL Server Instance. The

maximum that you can use is **800** seconds. If the value is set to **0**, the option is disabled.

- **Instance Name:** Type the name of the instance to configure. If it is a named instance, type the full instance name.

6 To save the settings, click **OK**.

Entering default logon credentials for a specific SQL Server Instance

If a specific instance uses different logon credentials, such as those described in Authentication Modes, from the credentials used by default, use this procedure to enter the different credentials. Information that you enter during this process *overrides* what you specified in Entering logon credentials for all instances on a client.

1 In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.

2 In the selection tree, open the applicable client node.

The client node can be a standalone server or Virtual Client.

3 Double-click **Plug-in *for SQL Server***, and double-click the **All Instances** node to display the list of SQL Server Instances.

Because you can use a single SQL Server installation to create multiple instances and configure authentication differently for each instance, NetVault Backup supports the use of different authentication information for different instances. If you only create one instance, that node is labeled “(local).”

4 Select the node for the first SQL Server Instance or AlwaysOn Availability Group that you want to configure; if you only have one instance, select the **“(local)”** node.

5 From the context menu, select **Configure**.

6 Complete the applicable fields:

- **Logon security mode:** Based on the ***Authentication Mode*** set for the SQL Server Instance, you can specify either a Windows Administrator or a SQL Server Administrator account. To specify a Windows Administrator account, select **Windows**. For a SQL Server Administrator account, select **SQL Server**.

  IMPORTANT: You can only select **SQL Server** if the ***SQL Server Authentication Mode*** is set to ***Mixed Mode/SQL Authentication***.

- **Login Details:** Depending on the option selected in the **Logon security mode** section, provide the appropriate account information in this section:
  - For Windows authentication, provide the following information:
    - **Administrator User Name:** Specify either a Local or Domain Windows Administrator user name.
    - **Password:** Enter the password associated with the user specified in the preceding field.
    - **Windows Domain:** If a Domain Administrator is specified in the **Administrator User Name** field, enter the name of the domain. You can leave this field blank if a Local Administrator user name is specified.
  - For SQL Server authentication, provide the following information:
    - **Administrator User Name:** Specify a SQL Server user with the sysadmin role.
    - **Password:** Enter the password associated with the user specified in the preceding field.

- **Instance Name:** Usually, the plug-in automatically identifies and completes this field, and you cannot change it. If the plug-in is unable to determine the instance name, such as when you are configuring an AlwaysOn Availability Group, specify the exact name of the SQL Server Instance running. When NetVault Backup locates the instance, the instance is displayed on the **NetVault Backup Selections** page for browsing and inclusion in a backup job.

7 If you are configuring Virtual Clients for an AlwaysOn Availability Group and you anticipate having to create additional backup jobs or modify existing backup jobs on the secondary nodes, perform the following steps:

   a Fail over the primary node to the secondary node.

   b Repeat Step 1 through Step 6.

   c Fail back to the primary node.

8 To save the settings, click **OK**.

9 If you need to enter different authentication information for additional SQL Server Instances, repeat Step 4 through Step 8 until all instances are configured.

   With the account properly configured, you can click the **All Instances** node to display the specified instances.

10 If your environment uses multiple Virtual Servers in the same SQL Server Failover cluster, complete the following steps for each Virtual Client you created:

   a In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.

   b In the selection tree, open the newly created Virtual Client that was set up for the first SQL Server Virtual Server.

   c Double-click **Plug-in *for SQL Server***, and double-click the **All Instances** node.

   All the SQL Server Virtual Servers within the failover cluster are revealed.

   d Select the first SQL Server Virtual Server in the tree, and then select **Configure** from the context menu.

   e Complete the applicable fields.

   With the configuration set for the first SQL Server Virtual Server in its corresponding Virtual Client, repeat the process for ***all*** remaining SQL Server Virtual Servers.

   f Repeat steps Step b through Step e for all remaining Virtual Clients, ensuring that the proper SQL Server Virtual Server is selected for each Virtual Client.

Changing the TCP/IP port for SQL Server

The plug-in does not communicate with a specific port. The plug-in connects with a SQL Server driver, which then communicates with the Database Engine on the configured TCP/IP port. Because the plug-in does not directly use the TCP/IP port itself, you can configure the SQL Server Database Engine to monitor a different port without having to reconfigure the plug-in.

Setting up SQL Server Authentication for SQL Server 2012 or later

Starting with SQL Server 2012, security privileges changed and you cannot use the LocalSystem account. Previously, you could use the LocalSystem account as the default for the sysadmin role. For SQL Server 2012 or later, use a domain account—including Administrator—that has the SQL Server sysadmin role or change the **Log on as** option for SQL Server Service to use a domain user that has the required privileges. If the SQL Server is not part of a domain, you can use a local user that has the sysadmin role or change the **Log on as** option for SQL Server Service to use a local user that has the required privileges.

1. Ensure that the selected domain or local user account has the **sysadmin** role assigned to it.
2. Complete one of the following:
   - In **Windows Control Panel > Administrative > Services**, locate and select **NetVault Process Manager**, and click **Stop**. Right-click **NetVault Process Manager**, and select **Properties**. On the **Log On** tab, ensure that NetVault Backup runs using the account that has the SQL Server sysadmin role. Start the NetVault Process Manager.
   - In the **SQL Server Configuration Manager**, click **SQL Server Services**. In the details pane, right-click the name of the applicable SQL Server Instance, and click **Properties**. In the **SQL Server *<instanceName>* Properties** dialog box, click the **Log On** tab. For **Log on as**, select the account that has the SQL Server sysadmin role. In **Windows Control Panel > Administrative > Services**, stop and start the **SQL Server Service**.
   - Use **SQL Server Management Studio** to add the domain or local user account that has the sysadmin privileges to the SQL Server. You can use SQL Server Management Studio to add this account, or enter the following in a command prompt:

     ```
     CREATE LOGIN [<domainName>\<loginName>] FROM WINDOWS;
     GO
     ```

     For more information, see https://technet.microsoft.com/en-us/library/ms189751%28v=sql.110%29.aspx.

     ```
     SP_ADDSRVROLEMEMBER '<domainName>\<loginName>', 'sysadmin'
     GO
     ```

     For more information, see https://technet.microsoft.com/en-us/library/ms186320(v=sql.110).aspx.

NOTE: You can assign the SQL Server **sysadmin** role to the default Local System user (NT AUTHORITY\SYSTEM), which lets the Plug-in *for SQL Server* log into the SQL Server Instances without prompting for credentials. Provisioning the Local System user (NT AUTHORITY\SYSTEM) with the **sysadmin** role might result in other applications being able to log into the SQL Server Instances without providing credentials. Before assigning the **sysadmin** role to the Local System user (NT AUTHORITY\SYSTEM), ensure that it is allowed under your compliance requirements.

Configuring domain administrator account for service exploration

For certain activities on Windows, it is necessary to impersonate a member of Administrators group on the client(s). An **Enter Domain Administrator** option is added in the context menu to configure a Windows Domain Administrator for clusters, even if the default login configuration is not a Windows user.

1. In the Navigation pane, click **Create Backup Job**, and click **Create New** next to the **Selections** list.
2. In the selection tree, open the applicable client node.

   The client node can be a standalone server or Virtual Client.
3. Click **Plug-in *for SQL Server***.
4. From the context menu, select **Enter Domain Administrator**.
5. Enter the login credentials for a domain account which is a member of the Administrators group on all cluster nodes:
   - **User Name:** Specify a domain Windows Administrator user name.
   - **Password**: Enter the password associated with the user specified in the preceding field.
   - **Domain**: Enter the name of the domain.

     NOTE: The user whose login credentials are entered must be a member of the Administrators group on all nodes of a cluster, if the client is a virtual client.
   - **Reset to blank saved User Name, Password, and Domain**: Select this option to reset to the blank or empty, the values stored in the plug-in for User Name, Password, and Domain.
6. To save the settings, click **OK**.

5

Backing up data

Defining a backup strategy

The purpose of creating SQL Server backups is to recover a database that is damaged from media failure or data corruption. Reliable use of backup for recovery requires a strategy that maximizes data availability and minimizes data loss, while accounting for defined business requirements.

A backup strategy is divided into two pieces: a backup piece and a restore piece.

- The backup piece defines the type and frequency of backups that are required to meet the goals for availability of the database and for minimizing data loss.
- The restore piece defines who is responsible for performing restores, and how restores should be performed to recover from the particular type of damage or failure.

> **IMPORTANT:** If you use Plug-in *for SQL Server*, verify that you are not using a third-party program to complete additional backups of SQL Server. Otherwise, a conflict might occur during the restore or recovery process when the differential backups do not match.

For more information, review the following topics:

- Reviewing the available backup methods
- Reviewing SQL Server recovery models
- Defining an Online VDI backup strategy and reviewing types
- Examples of VDI online backup sequences
- Defining a VSS backup strategy and reviewing types
- Examples of VSS backup sequences
- Backing up system databases
- Backing up replicated databases

Reviewing the available backup methods

The plug-in offers the following backup methods:

- Online Virtual Device Interface (VDI)
- Volume Shadow Copy Service (VSS)

The plug-in supports implementing an Online VDI or a pure VSS backup strategy, not a combination of the two.

Online Virtual Device Interface (VDI)

Microsoft supports the ability to perform online backups that use the **VDI** Application Program Interface (API) to integrate with a range of backup devices supported by NetVault Backup. Online VDI backups provide maximum reliability and flexibility when defining a backup strategy. This method supports the full range of backup types and options available with SQL Server's Transact SQL language along with the ability to handle several recovery scenarios.

The backup-and-restore strategies available with the Online through VDI Backup Method are thoroughly discussed in the *Backing Up and Restoring Databases* section of the *SQL Server Books Online*.

Plug-in *for SQL Server*'s Online through VDI backup method is the preferred backup method.

In VDI backup method, Plug-in *for SQL Server* creates **N** (where, **N** represents the number of databases) number of user connections with SQL Server. Maximum one user connection is active at a time.

Database name length limit for VDI backup method

NetVault Backup supports VDI backup for databases with names that do not exceed 117 characters in length.

To perform VDI backups, NetVault Backup Plug-in *for SQL Server*, use the BACKUP DATABASE Transact-SQL command. The Plug-in *for SQL Server* uses the BACKUP DATABASE command that includes the NAME clause with a limit of 128 characters. The Plug-in *for SQL Server* constructs the value to be passed to the NAME clause, using the database name, and a timestamp. Given the length in characters of the timestamp, results in 117 characters in length available for the name of the database.

If you are performing VDI backups including databases with names exceeding 117 characters in length, the backups complete with warnings, or with failed status. The NetVault Backup binary log shows the following ODBC message:

```
ODBC error: [Microsoft][ODBC SQL Server Driver][SQL Server]Could not insert a backup or restore history/detail record in the msdb database. This may indicate a problem with the msdb database. The backup/restore operation was still successful.
```

If you are performing backups using the VDI backup method, Quest recommends the name of the databases in your environment, to not exceed 117 characters. This limitation does not apply to backups performed using the VSS backup method.

Volume Shadow Copy Service (VSS)

Microsoft supports the ability to create snapshots of SQL Server data using VSS. VSS allows volume backups to be performed while applications on the system continue to write to the volumes. Microsoft provides a **SQL Server VSS Writer** that permits backup programs such as Plug-in *for SQL Server* to copy SQL Server data while SQL Server is running. VSS-based backups do not negatively affect SQL Server's performance or stability.

In VSS backup method, Plug-in *for SQL Server* creates **2*N** (where, **N** represents the number of databases) number of user connections with SQL Server. Maximum **N** user connections are active at a time.

IMPORTANT: A disadvantage of VSS backups is that the SQL Server VSS Writer does not support the ability to restore a database to an alternate SQL Server Instance. This disadvantage applies whether that instance is on the same server and host or an alternate server and host.

Using VSS, you can:

- Use snapshots to perform consistent backups to disk- or tape-based storage devices.
- Create and store snapshots as backups on NetVault Backup-supported disk arrays.

> **NOTE:** The **Backup Files to Storage** option is supported with any disk-based storage. To use the **Retain Snapshot as Persistent** and **Discard After** options, the data that you back up must reside on a NetVault Backup-supported disk array. Also, for persistent snapshots, only the metadata is copied to the target.

Reviewing SQL Server recovery models

When a database is created, a **recovery model** is enabled. Microsoft defines a recovery model as a “database property that controls the basic behavior of backup and recovery of the database.” The database’s recovery model controls how its transactions are logged, whether the transaction log can be backed up, and which kinds of restores are supported. SQL Server provides three different recovery models: Simple, Full, and Bulk-Logged.

- **Simple Recovery Model:** With a Simple Recovery Model, log backups are not supported. Therefore, changes since the most recent backup are not protected. In the unfortunate event of failure, these changes must be re-run. PIT recovery is not allowed.
- **Full Recovery Model:** Full Recovery Model databases require log backups; therefore, no work is lost due to a lost or damaged data file. PIT recovery is supported, assuming backups are complete up to the point-of-failure.
- **Bulk-Logged Recovery Model:** Bulk-Logged Recovery Model databases require log backups. The Bulk-Logged Recovery Model is a variation of the Full Recovery Model that permits high-performance bulk-copy operations. This model reduces log space usage by bulk-logging most bulk operations. If a log is damaged or bulk-operations have occurred since the most recent Transaction Log backup, these changes must be re-run. PIT recovery is not supported for bulk-logged databases.

Choosing the best recovery model

Consider the following when choosing a recovery model for a database:

- **Simple Recovery Model:** The Simple Recovery Model should only be enabled for databases that are not updated frequently such as test, development, or databases mostly containing read-only data.
- **Full Recovery Model:** The Full Recovery Model should be enabled for transactional databases where full recoverability and preventing work loss in a full range of recovery scenarios is required.
- **Bulk-Logged Recovery Model:** The Bulk-Logged Recovery Model should be used temporarily when bulk operations, such as bulk inserts or index creation, are performed on Full Recovery Model databases. The Bulk-Logged Recovery Model increases performance and reduces log space consumption during these operations; you can switch databases back to full recovery immediately after the bulk operations have completed.

For more information, see *Recovery Models and Transaction Log Management* in the *SQL Server Books Online*.

Defining an Online VDI backup strategy and reviewing types

After selecting the recovery model that meets your requirements for each database, you can design and implement a corresponding backup strategy. When defining a SQL Server Online VDI Backup strategy, answer the following questions:

- Is there a predictable off-peak period for full backups?
- What is the frequency of updates and changes?
- Are changes confined to a small or large number of tables in a database?

Answering these questions helps you define the type and frequency of backups that should be implemented.

The plug-in provides the following types of Online VDI Backup:

- Full Database backup for Online VDI
- Differential Database backup for Online VDI
- Copy-Only backup for Online VDI
- Incremental Transaction Log backup for Online VDI
- Tail-Log backup for Online VDI
- Full File and Filegroup backup for Online VDI
- Differential File and Filegroup backup for Online VDI
- Partial Database backup for Online VDI
- Differential Partial Database Backup for Online VDI

Full Database backup for Online VDI

Full Database backups are supported by:

- **SQL Server versions:** All
- **Recovery models:** All

A Full Database backup is a backup of the entire database. It also includes part of the transaction log, which enables recovery of the database to the point at which the backup was completed.

Full Database backups consume more space and time per backup and are typically supplemented by differential backups, which are created more frequently. With Full Database backups, you can re-create an entire database in one step by restoring the database.

Differential Database backup for Online VDI

Differential Database backups are supported by:

- **SQL Server versions:** All
- **Recovery models:** All

With a Differential Database, back up only the data that has changed since the last Full Database backup is backed up. Differential backups are smaller and faster to create than the full backups.

A Differential Database backup is useful if some of the database's tables are modified more frequently than others. In this case, Differential Database backups allow you to back up frequently without the overhead of Full Database backups.

Copy-Only backup for Online VDI

Copy-Only backups are supported by:

- **SQL Server versions:** 2008 and later
- **Recovery models:** All

Copy-Only backups are independent of the normal sequence of backups. A Copy-Only backup cannot serve as the base backup for a series of differential backups as a Full Backup can. Performing a Copy-Only backup does not affect what is backed up by the next differential backup. Copy-Only backups are ideal for situations that require special purposes, such as creating test environments or standby databases.

Incremental Transaction Log backup for Online VDI

Incremental Transaction Log backups are supported by:

- **SQL Server versions:** All
- **Recovery Models:** Full or Bulk-Logged Only

SQL Server Transaction Logs are essential to data recovery and must be backed up regularly. With Transaction Log backups, you can recover the database to a point-of-failure or specific point.

An Incremental Transaction Log backup captures all the transaction logs including those logs generated since the last Full Database/File or Filegroup, Differential Database/File or Filegroup, or Incremental Transaction Log backup. An Incremental Transaction Log backup should not be performed in the following conditions:

- Until a Full Database or Full File and Filegroup backup has been created because the transaction log contains changes made to the database after the last backup was created.
- After the transaction log has been manually truncated, until a Full Database or Differential Database backup has been performed. Microsoft strongly recommends against manually truncating transaction logs.

Tail-Log backup for Online VDI

Tail-Log backups are supported by:

- **SQL Server versions:** 2008 and later
- **Recovery models:** Full Only

A Tail-Log backup captures the transaction logs that have not yet been backed up and is the last backup restored in a restore sequence. SQL Server requires you to back up the tail of the log before restoring a database that is attached to the SQL Server Instance. Therefore, a Tail-Log backup is the first step in most restore sequences.

A Tail-Log differs from an Incremental Transaction Log in the following ways:

- You can try to run a Tail-Log backup even if the database does not start, for example if the database is damaged or is offline. If the database is damaged, a Tail-Log backup succeeds only if the log files are undamaged, the database is in a state that supports Tail-Log backups, and the database does not contain any bulk-logged changes.
- A Tail-Log might contain incomplete metadata if the database is damaged because some of the metadata normally available for log backups may be unavailable in a Tail-Log backup. However, the captured log is complete and usable.

  A Tail-Log's data might not be fully self-contained if the database is not online and undamaged at the time of the Tail-Log backup. Transactional data from a Tail-Log is always complete and usable, but if the Tail-Log backup is initiated while the database is damaged or offline, metadata might be only partially captured. If this issue occurs, a recovery from the Tail-Log leaves the **“backupfilegroup”** table missing some information about filegroups, and the “**has_incomplete_metadata”** column of the **“backupset”** table is set to 1.

- **WARNING: Because Tail-Log backups place the databases in Offline Mode, only perform a Tail-Log backup if you are preparing to perform a restore and recovery. Tail-Log backups should not be used as a substitute for regular or unplanned Incremental Transaction Log backups.**

- If you are working with an AlwaysOn Availability Group, use SQL Server Management Studio to remove the primary and secondary databases from the group before completing a Tail-Log backup. After the databases are removed from the group, their status changes to Restoring, and they are owned by the SQL Server Instance assigned the Primary Role in the group. To complete the Tail-Log backup, select the databases under the SQL Server Instance.

Full File and Filegroup backup for Online VDI

Full File and Filegroup backups are supported by:

- **SQL Server versions:** All
- **Recovery models:** All

A Full File and Filegroup backup backs up all the data in one or more files or filegroups. A complete set of Full File and Filegroup backups is equivalent to a Full Database backup.

Full File and Filegroup backups have the following advantages over Full Database backups:

- With a Full File and Filegroup backup, you can back up and restore the files in a database individually. This process can speed up the recovery process as you can restore only damaged files without restoring the rest of the database.
- Full File and Filegroup backups increase the flexibility in dealing with large databases that contain data with varying update characteristics. For example, consider the following recommendations:
  - Back up frequently modified data often.
  - Back up infrequently modified data less often.
  - Back up read-only data once.

The primary disadvantage of Full File and Filegroup backups is the risk that a media failure can render an entire database unrecoverable if a damaged file lacks a backup.

By default, file backups contain enough log records to roll forward the file to the end of the backup operation. Under Simple Recovery Model, the Full File and Filegroup backups are restricted to read-only secondary filegroups. You can create a file backup of a read/write filegroup, but before you can restore the read/write file backup, you must set the filegroup to read-only and take a differential read-only file backup.

Differential File and Filegroup backup for Online VDI

Differential File and Filegroup backups are supported by:

- **SQL Server versions:** All
- **Recovery models:** All

A Differential File and Filegroup backup captures only the data changed since the last file backup. These backups are fast because the Database Engine tracks changes made since the file was last backed up, eliminating the need to scan the file.

For Simple Recovery Model databases, Differential File backups provide a faster, space-saving way to create backups of current files. Differential File backups also reduce recovery time for Full Recovery Model databases by reducing the number of transaction logs that must be restored.

Differential File and Filegroup backups should be considered when:

- You are backing up some files much less frequently than others.
- Your files are large and the data is updated infrequently, or the same data is updated repeatedly.

Partial Database backup for Online VDI

Partial Database backups are supported by:

- **SQL Server versions:** 2008 and later
- **Recovery models:** Simple

Designed to create flexibility for Simple Recovery Model databases, a Partial Database backup is similar to a Full Database backup except that a Partial does not contain all the database's filegroups. A Partial Database backup contains all the data in the primary filegroup and every read/write filegroup. Optionally, specified read-only files can also be included. However, a partial backup of a read-only database contains only the primary filegroup.

A Partial Database backup is useful when a database contains read-only files that you do not need to include in every database backup. A Partial Database backup can also be used as a basis for a piecemeal restore scenario, which the plug-in does not support.

Differential Partial Database Backup for Online VDI

Differential Partial Database backups are supported by:

- **SQL Server versions:** 2008 and later
- **Recovery Models:** Simple

Used only with Partial Database backups, a Differential Partial backup contains only those extents changed in the primary filegroup and read/write filegroups since the previous partial backup. If only some of the data captured by a partial backup has changed, using a Differential Partial backup lets you back up frequently without the overhead of Full Database backups.

For more information on the effect of adding, deleting, or modifying Filegroups on Partial Differential backups, see *Differential Partial Backups* in the *SQL Server Books Online*.

Examples of VDI online backup sequences

Following are a few examples of backup sequences that support multiple restore scenarios and that you can implement to meet your SQL Server data protection requirements.

- Simple recovery model backup sequences
- Full recovery model backup sequences

Simple recovery model backup sequences

- **Full Database backups only:** When requirements guarantee data protection up to the previous day, performing Full Database backups nightly should be sufficient when one or more of the following conditions exist:
    - Backup windows are large.
    - Databases are small.
    - Updates are infrequent across entire database.
    - Databases are only for testing or development purposes.
    - Entire database is read-only.
- **Full Database and Differential Database backups:** When requirements guarantee data protection up to the previous day and backups must be completed quickly, Full Database backups coupled with Differential Database backups is a valid strategy. For example, Full Database backups are performed every Sunday night at 11:00 P.M. while Differential Database backups are performed Monday through Saturday at 11:00 P.M. Each Differential Database backup includes all the changes since the last Full Database backup, which is known as the differential base.

No matter when recovery is performed, the same number of restore jobs is required. For example, if recovery is performed on Tuesday, Sunday's Full Database backup and Monday's Differential Database must be restored. Whereas, if recovery is performed on Thursday, Sunday's Full Database backup followed by Wednesday's Differential Database backup must be restored.

Even though Differential Database backups increase not only in size but in duration, restores are quicker due to the fewer number of restore jobs that must be run.

- **Full File and Filegroup backups and Partial Database backups:** When a Simple Recovery Model database includes secondary read-only filegroups, a valid strategy includes Partial Database backups for the primary and read/write filegroups and Full File and Filegroup backups for read-only secondary filegroups. This strategy ensures recoverability of the entire database without requiring backup of the read-only filegroups every time the read/write filegroups are backed up.

  Secondary read-only filegroups only need to be backed up once after the initial load and after subsequent updates. When requirements guarantee data protection up to the previous day for read/write filegroups, performing nightly Partial Database backups should be sufficient.

- **Full File and Filegroup backups and Partial Database and Differential Partial Database backups:** When requirements guarantee data protection for read/write data up to the previous day and backups must be completed quickly, Partial Database backups coupled with Differential Database Backups for the read/write data and Full File and Filegroup backup for the secondary read-only filegroups is a valid strategy.

  For example, Full File and Filegroup backups are performed for read-only data only after the initial load and subsequent updates. Partial Database backups, which include the primary filegroup and all read/write filegroups, are performed every Sunday night at 11:00 P.M. In addition, Differential Partial Database backups for the same read/write filegroups are performed Monday through Saturday at 11:00 P.M. Each Differential Partial Database backup includes all the changes since the last Partial Database Backup, which is known as the differential base.

  No matter when recovery is performed, the same number of restore jobs is required. For example, if recovery is performed on Tuesday, Sunday's Partial Database backup and Monday's Differential Partial Database must be restored. Whereas, if recovery is performed on Thursday, Sunday's Partial Database backup followed by Wednesday's Differential Partial Database backup must be restored.

  Even though Differential Partial Database backups increase not only in size but also in duration, restores are quicker due to the fewer number of restore jobs that must be run.

Full recovery model backup sequences

- **Full Database and Incremental Transaction Log backups:** When requirements mandate PIT data protection, performing Full Database backup nightly plus Incremental Transaction Log backups every four to eight hours should be sufficient when one or more of the following conditions exist:
  - Backup windows are large.
  - Databases are small.
  - Updates are infrequent across the entire database.

  For example, Full Database backups are performed every night while Incremental Transaction Log backups are performed every four to eight hours. Each Incremental Transaction Log backup includes the transaction logs since the last Full Database or Incremental Transaction Log backup.

  Restore sequences that use Incremental Transaction Log backups require that every Incremental Transaction Log backup between the last Full Database backup and the point-of-failure is restored in succession. This process can lead to longer restores and increased intervention to initiate multiple restore jobs. To speed up restores, include Differential Database backups, which reduces the number of Incremental Transaction Log backups that must be restored.

- **Full Database and Differential Database and Incremental Transaction Log backups:** When requirements mandate PIT data protection and restores of the complete database must be completed quickly, Full Database plus Differential Database plus Incremental Transaction Log backups is an ideal strategy.

For example, Full Database backups are performed every Sunday night at 11:00 P.M. Differential Database backups are performed Monday through Saturday at 11:00 P.M. Incremental Transaction Log backups are performed every hour. Each Differential Database backup includes all the changes since the last Full Database backup, and each Incremental Transaction Log backup includes the transaction logs since the last Differential Database backup.

Restore sequences that use Incremental Transaction Log backups require that every Incremental Transaction Log backup between the last Full Database or Differential Database backup and the point-of-failure is restored in succession. This process can lead to longer restores and increased intervention to initiate multiple restore jobs. To speed up restores, increase the frequency of the Differential Database backups to every four to eight hours. This increase reduces the number of Incremental Transaction Log backups that must be restored. This strategy provides complete protection of the entire database with the simplest administration while supporting multiple restore scenarios.

- **Full File and Filegroup and Differential File and Filegroup and Incremental Transaction Log backups:** When requirements mandate PIT data protection and restores must be completed quickly, Full File and Filegroup plus Differential File and Filegroup plus Incremental Transaction Log backups is an ideal strategy. This strategy lets you restore a complete database or the files or filegroups that are damaged. By being able to restore only what is needed, restores are faster and data loss is minimized.

  For example, Full File and Filegroup backups are performed every Sunday night at 11:00 P.M. Differential File and Filegroup backups are performed Monday through Saturday at 11:00 P.M. Incremental Transaction Log backups are performed every hour. Each Differential File and Filegroup backup includes all changes since the last Full File and Filegroup backup, and each Incremental Transaction Log backup includes the transaction logs since the last Differential File and Filegroup backup.

  Restore sequences that use Incremental Transaction Log backups require that you restore every Incremental Transaction Log backup between the last Full File and Filegroup or Differential File and Filegroup backup and the point-of-failure in succession. This process can lead to longer restores and increased intervention to initiate multiple restore jobs. To speed up restores, increase the frequency of the Differential File and Filegroup backups to every four to eight hours. This change reduces the number of Incremental Transaction Log backups that must be restored. This strategy provides complete protection of the entire database with the flexibility to restore only the files or filegroups that are damaged while supporting most restore scenarios.

Defining a VSS backup strategy and reviewing types

Plug-in *for SQL Server* provides the following types of VSS backups:

- **Full Database backup:** Full Database backups are supported by:
  - **SQL Server versions:** 2008 and later
  - **Recovery models:** All

  A VSS Full Database backup backs up database data and all the log files necessary to bring the database up to a transactionally consistent state at restore time. Full Database backups consume more space and time per backup and are typically supplemented by differential backups, which are created more frequently. With Full Database backups, you can re-create an entire database in one step by restoring the database.
- **Differential Database backup:** Differential Database backups are supported by:
  - **SQL Server versions:** 2008 and later
  - **Recovery model:** All

  A VSS Differential Database backup backs up only the data that has changed since the most recent base Full Database backup. A Differential Database backup contains only those parts of the database files that have changed. Differential Database backups are smaller and faster to create than the Full Database backups. A Differential Database backup is useful if some of the database's tables are modified more frequently than others. In this case, Differential Database backups allow you to back up frequently without the overhead of Full Database backups.

> **IMPORTANT:** If a failover occurs in an Active/Passive Failover Cluster and your VSS-based backup strategy includes Differential Database backups, perform a Full Database backup of the modified configuration as soon as possible. When the plug-in detects that a failover occurred and a different node holds the active role, the plug-in automatically runs a full backup for the next scheduled backup. However, if you must perform a restore to the new active node *before* a new full backup is run, database changes that occurred between the last full backup and the new full backup might be lost.

- **Copy-Only backup:** Copy-Only backups are supported by:
  - **SQL Server versions:** 2008 and later
  - **Recovery models:** All

  VSS Copy-Only backups are independent of the normal sequence of backups. Unlike a full backup, a Copy-Only backup cannot serve as the base backup for a series of differential backups. Also, performing a Copy-Only backup does not affect what is backed up by the next differential backup. Copy-Only backups are ideal for situations that require special purposes, such as creating test databases.

Examples of VSS backup sequences

Following are few examples of backup sequences that support multiple restore scenarios and that you can implement to meet your SQL Server data protection requirements.

- **Full Database backups only:** When requirements guarantee data protection up to the previous day, performing Full Database backups nightly should be sufficient when one or more of the following conditions exist:
  - Backup windows are large.
  - Databases are small.
  - Updates are infrequent across entire database.
  - Databases are only for testing or development purposes.
  - Entire database is read-only.
- **Full Database and Differential Database backups:** When requirements guarantee data protection up to the previous day and backups must be completed quickly, Full Database backups coupled with Differential Database backups is a valid strategy. For example, Full Database backups are performed every Sunday night at 11:00 P.M., while Differential Database backups are performed Monday through Saturday at 11:00 P.M. Each Differential Database backup includes all the changes since the last Full Database Backup, which is known as the differential base.

  No matter when recovery is performed, the same number of restore jobs is required. For example, if recovery is performed on Tuesday, Sunday's Full Database backup and Monday's Differential Database backup must be restored. Whereas, if recovery is performed on Thursday, Sunday's Full Database backup followed by Wednesday's Differential Database backup must be restored.

  Even though Differential Database backups increase not only in size but in duration, restores are quicker due to the fewer number of restore jobs that must be run.
- **Copy-Only backups:** When requirements demand that a backup be created without interfering with regular scheduled backup sequences, Copy-Only backups is a valid strategy. Often times, you must create a full backup outside of the regular backup schedule, such as when refreshing your test databases. This process could potentially interfere with the regular backup schedule and throw off subsequent differential backups. The Copy-Only backup allows administrators to run non-regular Full Database backups at any time without interfering with scheduled backups.

Understanding snapshot-based backups

If you are using SQL Server 2010 or later, the plug-in can use a hardware or software VSS provider to create persistent or non-persistent VSS-based snapshots. The plug-in uses a VSS provider to create snapshots on the client, and then copies the selected data from the snapshot, or snapshots, to a storage device.

If you are using a standalone SQL Server deployment, you can use hardware-based, integrated VSS snapshots with the Dell Compellent storage array. (For AlwaysOn Availability Group environments, only VSS backups that use the software-based Microsoft VSS provider are supported.)

To create and use persistent snapshots, the data that you want to back up must reside on NetVault Backup-supported disk arrays. If you attempt to create persistent snapshots with data residing on an unsupported disk array, or with data residing in local drives (for which snapshots are taken using the Microsoft Software VSS Provider), taking the required hardware snapshots fails, and the plug-in defaults to using software snapshots. When taking software snapshots the data is only backed up to storage (even if the option **Backup Files to Storage** is not selected), and non-persistent snapshots are created.

The same issue occurs if your backup includes data that resides in different storage that mixes NetVault Backup-supported disk arrays and unsupported disk arrays or local drives.

IMPORTANT: For AlwaysOn Availability Group deployments, hardware snapshots are not currently supported by the plug-in. If you are running an AlwaysOn Availability Group environment, ensure that the software-based Microsoft VSS provider is in place on all SQL Servers included in the AlwaysOn Availability Group and ensure that the software-based provider has preference over any vendor's hardware-based VSS provider. Also be aware that NetVault Backup-managed persistent snapshots are not supported in AlwaysOn Availability Group deployments.

For more information about NetVault Backup-supported OS versions and disk arrays, see the *Quest NetVault Backup Compatibility Guide*.

Backing up system databases

The SQL Server system databases are essential for the operation of a server instance. These backups enable recovery of the SQL Server if a system failure occurs, such as loss of a hard disk.

The system databases that you must always back up include **msdb**, **master**, and **model**. The **master** and **msdb** databases have a Simple Recovery Model while the **model** database has a Full Recovery Model. For replicated databases, you should also back up the **distribution** database.

- **master:** The **master** database records system-level information for a SQL Server system. For the **master** database, SQL Server supports only **Full Database** backups. Quest recommends that you schedule routine Full Database backups of **master**.
- **msdb and model:** SQL Server uses the **msdb** database to store data. SQL Server uses the **model** database as a template when you create a user database. Back up the **model** and **msdb** databases whenever they are updated. You can back up the **model** and **msdb** databases in the same way that you back up user databases. Quest recommends that you create only Full Database backups of the **model** database. Because it is small and rarely changes, backing up the log is not required.

  NOTE: The plug-in cannot back up **tempdb** or **Resourcedb** databases.

For more information, see *Backing Up and Restoring System Databases* in *SQL Server Books Online*.

Backing up replicated databases

Database replication is supported from SQL Server 2008 onward. Plug-in *for SQL Server* supports all three types of replication: Transactional Replication, Merge Replication, and Snapshot Replication.

Replicated databases and their associated system databases should be backed up regularly, including the following:

- The **publication** database at the Publisher.
- The **distribution** database at the Distributor.
- The **subscription** database at each Subscriber.
- The **master** and **msdb** system databases at the Publisher, Distributor, and all Subscribers. These databases should be backed up at the same time as each other and the relevant replicated database.

  For example, back up the **master** and **msdb** databases at the Publisher at the same time that you back up the publication database. If the publication database is restored, ensure that the **master** and **msdb** databases are consistent with the publication database in terms of replication configuration and settings.

The replicated databases can be backed up in the same way as the non-replicated databases with the plug-in. If you perform regular Incremental Transaction Log backups, any replication-related changes should be captured in the Incremental Transaction Log backups. If you do not perform Incremental Transaction Log backups, a backup should be performed whenever a setting relevant to replication is changed.

For more information, see *Strategies for Backing Up and Restoring Snapshot and Transactional Replication, Strategies for Backing Up and Restoring Merge Replication*, and *Common Actions Requiring an Updated Backup* in the *SQL Server Books Online*.

Reviewing the compression features

Before configuring a backup, review the following information. This information applies to all versions of SQL Server that the plug-in supports, although some of the features described are not supported by all versions of SQL Server.

If your version of SQL Server supports it, you can use the SQL Server Backup Compression feature. The plug-in also provides three levels of compression, low, medium, and high, which you can use whether or not your server supports SQL Server Backup Compression. You can also turn off compression. The compression options include:

- **Use NetVault Backup Low Compression:** This method, which is selected by default, provides good compression with minimal processor usage. Quest recommends that you select this option when compression is needed but minimizing processing is more important.
- **Use NetVault Backup Medium Compression:** This method provides better compression but requires additional processor usage. Quest recommends that you select this option when improved compression is needed and additional processor usage during backup is not an issue.
- **Use NetVault Backup High Compression:** This method provides the best compression. Quest recommends that you select this option when compression is the most important factor and processor consumption is not an issue.
- **Use SQL Server Compression as Set in the SQL Server Instance:** This option is only available if the plug-in is connecting to a SQL Server Instance that supports the SQL Server Backup Compression feature. If SQL Server Backup Compression is enabled for this SQL Server Instance, the plug-in uses SQL Server Backup Compression for this instance. If it is disabled for this SQL Server Instance, the plug-in does not use SQL Server Backup Compression.
- **Use SQL Server Compression:** This option is only available if the plug-in is connecting to a SQL Server Instance that supports the SQL Server Backup Compression feature. Specifying this option tells the SQL Server to use a SQL Server Backup Compression algorithm to compress the data stored on backup media.

  For more information on SQL Server Backup Compression, see http://technet.microsoft.com/en-us/library/bb964719.aspx.

- **Do Not Use Compression:** To disable compression, select this option.

To view the compression ratio achieved after a backup job has run, click **Job Status** in the Navigation pane, and select the completed job, and click **View logs**. In the log that appears, locate a message that starts with "Compression Ratio." To view additional information, select the line, and click **More info**. You can then view the total number of bytes of uncompressed data that was backed up, the number of bytes that was sent to the backup media, and the number of administrative bytes included as a header, which is also included as part of the compressed data statistic but are extraneous to the actual compressed data. The compression ratio that is reported in the log is **[[(total uncompressed data) – (total compressed data)] x 100%] / (total uncompressed data)**. The number of header bytes is usually an insignificant part of the backup size, except when small databases are backed up using multiple NetVault Backup streams.

If you use the SQL Server Backup Compression feature, or if you choose not to use compression, the **VDI Streams when NetVault Backup Compression is not used** option is enabled.

If you use one of the NetVault Backup Compression methods, the **VDI Streams/Compression Threads** and **Number of Parallel NetVault Backup Streams** options are enabled. The VDI threads perform compression. During a backup, each VDI thread or connection accepts data from SQL Server, compresses it, and then writes it to the output device using a NetVault Backup stream. Because higher compression levels are more processor-intensive, consider making effective use of the number of processors available on the system running the NetVault Backup Client, that is, the system running the SQL Server Instance that you are backing up. For example, you might specify a number of VDI threads that is slightly less than the number of processors on your system. However, the number of backup streams should not exceed the number of output devices. For this reason, you must specify the number of VDI streams (**VDI Streams/Compression Threads**) and the number of NetVault Backup streams (**Number of Parallel NetVault Backup Streams**) separately. When you access the **SQL Server Backup Options** tab, the number of VDI streams initially displayed is one less than the number of processors detected on the NetVault Backup Client. The value specified in this field determines the number of VDI devices to be used for storing the backup job. In the default setting, all backup job options have ***one*** VDI device set to the job, when **Use Virtual Device Interface (VDI)** is selected. You can add additional VDI devices; however, the minimum number of VDI devices is ***1***, and the maximum number cannot exceed ***64***.

IMPORTANT: If you use the SQL Server Backup Compression feature or no compression, the number of VDI devices selected must be equal to or less than the number of available media or tape drives. For example, if the number of VDI devices selected is 5, a minimum of five media or tape drives must exist.

If you use NetVault Backup Compression, the number of parallel backup streams selected must be equal to or less than the number of available media or tape drives.

At the bottom of the **VDI Backup Options** section, the plug-in displays the number of processors on the NetVault Backup Client. If the processors use Hyper-Threading (HT), the number reflects it; a system with four physical processors and with HT enabled is identified as having eight processors. You can use this number when choosing the number of VDI streams if you use NetVault Backup Compression.

For optimal backup times when NetVault Backup Compression is used, base the entry for the **Number of Parallel NetVault Backup Streams** on the number of backup media. If your media uses striped disks, the fastest backup might occur with the number of backup streams set to the number of stripes. If the backup is writing to different hard drives that are not striped, with each NetVault Backup output device on a different hard drive, set the number of backup streams to the number of hard drives. If you are backing up to one or more VTLs on a single non-striped disk, use only one or two NetVault Backup streams. If you are backing up to a tape library with several tape drives, use no more backup streams, preferably fewer, than there are tape drives.

Whether or not you use NetVault Backup Compression, the number of VDI devices required for a restore is equal to the number that were used in the backup, and the number of backup streams required for the restore is also equal to the number that were used in the backup. If you use tape media and fewer tape drives are available for the restore than were used for the backup, the restore might fail. To avoid failure, Quest recommends that you use fewer NetVault Backup streams than you have tape drives available.

Performing Online VDI backups

An Online VDI backup using Plug-in *for SQL Server* includes the steps outlined in the following topics.

- Selecting data for an Online VDI backup
- Setting backup options for a VSS backup
- Finalizing and submitting an Online VDI backup job

Selecting data for an Online VDI backup

You must use sets—Backup Selection Set, Backup Options Set, Schedule Set, Target Set, and Advanced Options Set—to create a backup job.

Backup Selection Sets are essential for Incremental and Differential Backups. Create the Backup Selection Set during a Full Backup, and use it for Full, Incremental, and Differential Backups. The backup job reports an error if you do not use a Selection Set for the Incremental or Differential Backup. For more information, see the *Quest NetVault Backup Administrator's Guide*.

TIP: To use an existing set, click **Create Backup Job**, and select the set from the **Selections** list.

1. In the Navigation pane, click **Create Backup Job**.

   You can also start the wizard from the Guided Configuration link. In the Navigation pane, click **Guided Configuration**. On the **NetVault Configuration Wizard** page, click **Create backup jobs**.

2. In **Job Name**, specify a name for the job.

   Assign a descriptive name that lets you easily identify the job when monitoring its progress or restoring data. The job name can contain alphanumeric and non-alphanumeric characters, but it cannot contain non-Latin characters. On Windows, there is no length restriction; however, a maximum of 40 characters is recommended.

3. Next to the **Selections** list, click **Create New**.

4. In the list of plug-ins, open **Plug-in *for SQL Server***, and then open the **All Instances** node to view the individual instances.

   With the instances visible, you can do one of the following:

   - Select an entire instance for inclusion.

   IMPORTANT: If you are backing up data that was set up for use in a Virtual Client, select the SQL Server Virtual Server, or the items contained within, for inclusion in the backup. If you are using multiple SQL Server Virtual Servers in the same cluster, select the correct SQL Server Virtual Server whose data you want to back up. The instance name displayed on this page is the SQL Server Virtual Server that was established as the Virtual Client during installation.

   Check that you run the process using the Virtual Client and *not* from one of the nodes. If you open or expand one of the nodes and drill down through the hierarchy, you might see a dimmed SQL Server Instance identified as **local**. While the system might use this instance in maintaining log information, do not run any processes at this level.

   - Open an instance to display its individual databases, and select the databases to include in the backup.
   - Open an individual database to display its files and filegroups, and select the items to include in the backup.

> **IMPORTANT:** For the backup types Full File and Filegroup, Differential File and Filegroup, Partial Database, and Differential Partial Database, a read-only filegroup must be explicitly selected in the selection tree to include it in the backup. Though a green check mark appears against the read-only filegroups when you select the parent-level database, these filegroups are not backed up. When including a read-only filegroup in a backup, select the individual items instead of the parent-level database.

5 Click **Save**, enter a name in the **Create New Set** dialog box, and click **Save**.

The name can contain alphanumeric and non-alphanumeric characters, but it cannot contain non-Latin characters. On Linux, the name can have a maximum of 200 characters. On Windows, there is no length restriction. However, a maximum of 40 characters is recommended on all platforms.

Setting backup options for an Online VDI backup with Plug-in *for SQL Server*

The next step involves creating the Backup Options Set or selecting an existing one.

> **TIP:** To use an existing set, in the **Plugin Options** list, select the set that you want to use.

1 Next to the **Plugin Options** list, click **Create New**.

2 In the **Backup Method** section on the **SQL Server Backup Options** tab, select **Use Virtual Device Interface (VDI)**.

The VDI backup method lets you take advantage of Microsoft's VDI API, which provides maximum reliability and flexibility when defining a backup strategy.

3 Set the following options, if applicable:

- **Block Size (Bytes):** This field allows for the input of a device block size, in bytes. The default value is 64KB, that is, 65536 bytes.
- **Transfer Multiple:** The value specified in this field is used as a multiplier for the value specified in the **Block Size** field. The overall value of the two serves as the maximum allowable data transfer size; that is, the **Block Size** default of 65536 bytes multiplied by the default **Transfer Multiple** of 24, results in a maximum allowable transfer size of 1.6MB.

4 In the **Backup Type** section, select the applicable option:

- **Full Database**
- **Differential Database**
- **Copy-Only**
- **Incremental Transaction Log**
- **Tail-Log**
- **Full File and Filegroup**
- **Differential File and Filegroup**
- **Partial Database**
- **Differential Partial Database**

For information on these backup types, see Defining an Online VDI backup strategy and reviewing types.

5 In the **Incomplete Backup of ALL Items Selected** section, select the applicable option to instruct the plug-in on what it should do if this error condition occurs:

- **Complete with Warnings — Saveset Retained:** The job returns a status of **"Backup Completed with warnings"** and a backup saveset is created that includes the items that were successfully backed up.

- **Complete without Warnings — Saveset Retained:** The job completes and returns a status of **"Backup Completed."** The errors are logged in the NetVault Backup binary logs and ignored on the **Job Status** page. A backup saveset is created that includes the items that were backed up.
- **Fail — Saveset Retained:** The job returns a status of **"Backup Failed."** However, a backup saveset is generated that includes the items that were successfully backed up.
- **Fail — No Saveset Retained:** The job returns a status of **"Backup Failed"** and no saveset of backed-up objects is kept. That is, even if some of the objects were successfully backed up, the saveset is discarded.

If either of the following occurs, the plug-in overrides the **Incomplete Backup of ALL Items Selected** setting and responds as indicated:

- If a fatal error occurs, the job returns a **"Backup Failed"** status.
- If a Partial Database or Differential Partial Database backup is performed on a database whose backup target contains a read-only filegroup, the job returns a **"Backup Completed with warnings"** status.

6 In the **VDI Backup Options** section, select the applicable **Backup Compression** option; **Use NetVault Backup Low Compression** is the default:

- **Use NetVault Backup Low Compression**
- **Use NetVault Backup Medium Compression**
- **Use NetVault Backup High Compression**
- **Use SQL Server Compression as Set in the SQL Server Instance**
- **Use SQL Server Compression**
- **Do Not Use Compression**

7 If you selected one of the NetVault Backup Compression options, complete the **VDI Streams/Compression Threads** and **Number of Parallel NetVault Backup Streams** options, and then skip to Step 10.

The number of parallel backup streams selected *must* be equal to or less than the number of available media or tape drives.

8 If you selected either of the options related to the SQL Server Backup Compression feature, enter the number of VDI streams in the **VDI Streams when NetVault Backup Compression is not used** field, and then skip to Step 10.

The number of VDI devices selected *must* be equal to or less than the number of available media or tape drives.

9 If you selected **Do Not Use Compression**, enter the number of VDI streams in the **VDI Streams when NetVault Backup Compression is not used** field.

The number of VDI devices selected *must* be equal to or less than the number of available media or tape drives.

10 If you selected a **Backup Type** of **Differential Database** or **Incremental Transaction Log**, and the **Instance Node** is selected, use the **New Database Found in Differential/Incremental Backup** option to specify what action to take if a new database is found:

- **Do full database backup:** This option instructs the plug-in to perform a full backup of all databases added since the last **Full Database** backup.
- **Ignore:** Selecting this option instructs the plug-in to ***ignore*** any databases created since the last **Full Database** backup; that is, any databases that were created after the last backup are omitted from the **Differential Database** or **Incremental Transaction Log** backup.

11 If you selected a **Backup Type** of **Incremental Transaction Log** or **Tail-Log**, use the **Transaction Log Options** option to specify whether the inactive portion of the log must be truncated:

- **Normal:** Select this option if you want the plug-in to truncate the inactive portion of the log file and make it available for re-use. This option is the default for **Incremental Transaction Log** backups.

- **No Truncate:** Select this option to avoid truncation of the log during backup. This option must be selected for performing **Tail-Log** backups.

IMPORTANT: When the **Normal** option is enabled, SQL Server automatically truncates the Transaction Logs after sending the backup Transaction Logs to the plug-in. When the **No Truncate** option is enabled, the Transaction Logs are not truncated. If the **Normal** option is used and the *backup dies unexpectedly*, the Transaction Logs are lost permanently. If the plug-in fails to write the index or dies, the Transaction Logs cannot be restored and those transactions have already been deleted from the SQL Server logs. This issue results in errors when attempting to restore Transaction Log backups that were created after the failed backup job. If this situation occurs, make a full backup of your database to eliminate the need for these logs during restoration. If the backup encounters normal errors, the Transaction Logs are not lost.

12 If you want the plug-in to perform checksum on the backups, select the **Enable Backup Checksum** option, and then use **Error Encountered During Checksum** to indicate what the plug-in should do if it encounters an error.

Selecting this option tells SQL Server to verify the page, checksum or torn page, if this information is present on the page before writing a page to the backup media. Regardless of whether page checksums are present, SQL Server generates a separate backup checksum for the backup streams. Restore operations can optionally use the backup checksum to validate that the backup is not corrupt. The backup checksum is stored on the backup media, not on the database pages and can affect workload and backup throughput.

- **Continue after error:** With this default value selected, the backup job continues even if it encounters an invalid checksum.
- **Stop on error:** With this value selected, the backup job fails.

IMPORTANT: To change the default action for this option for all subsequent jobs, see Configuring plug-in options.

13 If you are backing up an AlwaysOn Availability Group, click the **AlwaysOn Availability Groups** tab, and select the applicable **Backup Replica Selection Algorithm** option:

NOTE: NetVault Backup considers all secondary replicas to have equal priority. If you choose an option that lets NetVault Backup select the replica to use for the backup process, NetVault Backup assigns preference to a replica that is functioning in synchronous-commit mode.

- **Primary Only:** To require NetVault Backup to use the primary replica, select this option.
- **Prefer Secondary:** To instruct NetVault Backup to give preference to using a secondary replica or the primary replica if a secondary replica is not available, select this option.
- **Any Replica:** To let NetVault Backup determine which replica, primary or secondary, to use, select this option.
- **Secondary Only (For Copy-Only Backup):** To require NetVault Backup to use a secondary replica, select this option.

  IMPORTANT: To back up a Secondary Copy, select the **Backup Type** of **Copy-Only**. If you select a different Backup Type, the job fails.

  If you back up a secondary replica in an AlwaysOn Group, SQL Server only lets you perform VDI Copy-Only backups. Only backups of the primary replica are supported with VSS.
- **As Configured in SQL Server:** To use the priority level set in your SQL Server configuration using SQL Server Management Studio, select this option.

14 Click **Save**, specify a name for the set in the **Create New Set** dialog box, and click **Save**.

The name can contain alphanumeric and non-alphanumeric characters, but it cannot contain non-Latin characters. On Windows, there is no length restriction; however, a maximum of 40 characters is recommended.

Finalizing and submitting an Online VDI backup job

The final steps include setting additional options on the Schedule, Target Storage, and Advanced Options pages, submitting the job, and monitoring the progress by using the Job Status and View Logs pages. These pages and options are common to all NetVault Backup Plug-ins. For more information, see the *Quest NetVault Backup Administrator's Guide*.

1. Use the **Schedule**, **Target Storage**, and **Advanced Options** lists to configure any additional required options.
2. Click **Save** or **Save & Submit**, whichever is applicable.

   TIP: To run a job that you have already created and saved, select **Manage Job Definitions** in the Navigation pane, select the applicable job, and click **Run Now**.

   You can monitor progress on the **Job Status** page and view the logs on the **View Logs** page. For more information, see the *Quest NetVault Backup Administrator's Guide*.

Performing VSS backups in SQL Server

Microsoft supports the ability to create snapshots of SQL Server data using VSS. VSS allows volume backups to be performed while applications on the system continue to write to the volumes.

Microsoft provides a **SQL Server VSS Writer** that permits backup solutions such as Plug-in *for SQL Server* to copy SQL Server data while SQL Server is running. VSS-based backups do not negatively affect SQL Server's performance or stability.

A VSS backup using Plug-in *for SQL Server* includes the steps outlined in the following topics.

- Selecting data for a VSS backup
- Setting backup options for a VSS backup
- Finalizing and submitting a VSS backup job

IMPORTANT: Before initiating VSS backups, verify that the **SQL Server VSS Writer** service is running on the SQL Server. The **SQL Server VSS Writer** is automatically installed with SQL Server. However, the service is not enabled to start automatically. For VSS backups, start this service from the Windows Services panel—**Start > Control Panel > Administrative Tools > Services**. Quest recommends that you set the Startup Type for this service to **Automatic**.

You can define a VSS backup job regardless of the state of this service. However, the job fails if it is run while the SQL Server VSS Writer service is not running. The log for the failed backup job contains a message stating "Please ensure the SQL Server Writer service is started."

Selecting data for a VSS backup

You must use sets—Backup Selection Set, Backup Options Set, Schedule Set, Target Set, and Advanced Options Set—to create a backup job.

Backup Selection Sets are essential for Incremental and Differential Backups. Create the Backup Selection Set during a Full Backup, and use it for Full, Incremental, and Differential Backups. The backup job reports an error if you do not use a Selection Set for the Incremental or Differential Backup. For more information, see the *Quest NetVault Backup Administrator's Guide*.

TIP: To use an existing set, click **Create Backup Job**, and select the set from the **Selections** list.

1. In the Navigation pane, click **Create Backup Job**.

   You can also start the wizard from the Guided Configuration link. In the Navigation pane, click **Guided Configuration**. On the **NetVault Configuration Wizard** page, click **Create backup jobs**.
2. In **Job Name**, specify a name for the job.

   Assign a descriptive name that lets you easily identify the job when monitoring its progress or restoring data. The job name can contain alphanumeric and non-alphanumeric characters, but it cannot contain non-Latin characters. On Windows, there is no length restriction; however, a maximum of 40 characters is recommended.
3. Next to the **Selections** list, click **Create New**.
4. In the list of plug-ins, open **Plug-in *for SQL Server***, and then open the **All Instances** node to view the individual instances.

   With the instances visible, you can do either of the following:

   - Select an entire instance for inclusion.
   - Open an instance to display its individual databases, and select the databases to include in the backup.
5. Click **Save**, enter a name in the **Create New Set** dialog box, and click **Save**.

   The name can contain alphanumeric and non-alphanumeric characters, but it cannot contain non-Latin characters. On Linux, the name can have a maximum of 200 characters. On Windows, there is no length restriction. However, a maximum of 40 characters is recommended on all platforms.

Setting backup options for a VSS backup

The next step involves creating the Backup Options Set or selecting an existing one.

TIP: To use an existing set, in the **Plugin Options** list, select the set that you want to use.

NOTE: During a VSS backup, the plug-in ignores all new databases and performs only the differential backup of databases that have had a full VSS backup first. An error message indicating that this issue has occurred displays in the NetVault Backup log.

1. Next to the **Plugin Options** list, click **Create New**.
2. In the **Backup Method** section on the **SQL Server Backup Options** tab, select **Use Volume Shadow Copy Services (VSS)**.
3. In the **Backup type** section, select the applicable option:

   - **Full Database**
   - **Differential Database**
   - **Copy-Only**

For information on these backup types, see Defining a VSS backup strategy and reviewing types.

4 In the **Snapshot Options** section, complete the following:

- **Backup Files to Storage:** This option, which is selected by default, copies the selected data from snapshots to a storage device. The plug-in uses a VSS provider to create persistent or non-persistent snapshots on the client, and copies the selected data from the snapshot, or snapshots, to the storage device.
- **Retain Snapshot as Persistent**: To retain persistent snapshots on a disk array, select this check box.
  - If the **Backup Files to Storage** and **Retain Snapshot as Persistent** check boxes are both selected, the plug-in copies the selected files to a storage device and adds the snapshot information to the backup index.
  - If you clear the **Backup Files to Storage** check box and select this check box, the plug-in writes only the index entries to the backup stream and creates persistent snapshots on the disk array.
- **Discard After:** If you selected the **Retain Snapshot as Persistent** check box and you want to delete persistent snapshots from a disk array after a specified period, select this check box and complete the expiration fields.

  Regardless of the expiration settings, a snapshot is automatically deleted when the associated saveset is retired. Therefore, this option is only useful when you want to discard a snapshot *before* the associated saveset is retired. To expire a snapshot immediately, you must log in to the storage array, and manually expire the snapshot. Otherwise, the snapshot expires according to the retention policy selected while creating the snapshot.

> **NOTE:** To use the **Retain Snapshot as Persistent** and **Discard After** options, the SQL Server that you back up must reside on a NetVault Backup-supported disk array and the vendor's hardware-based VSS provider must be in place. For more information, see Understanding snapshot-based backups.

5 In the **VSS Backup Options** section, select **Use multiple snapshots to limit demands on system resources**.

> **NOTE:** You must select the option **Use multiple snapshots to limit demands on system resources** for VSS Differential Database backup type if you selected it for VSS Full Database backup type.
>
> This option can be applied when the VSS backup method is selected. When a VSS snapshot is taken, the amount of system resources required increases with the amount of SQL Server databases that is included in the snapshot. Select this option to limit the amount of system resources that are used by the VSS snapshot. When this option is selected, the Plug-in *for SQL Server* groups the selected databases in groups or batches, and takes a snapshot for each batch, instead of including all the selected databases in one single snapshot. Hence, system resources are not exhausted.
>
> The option **Use multiple snapshots to limit demands on system resources** cannot be combined with the option **Retain Snapshot as Persistent**. If **Use multiple snapshots to limit demands on system resources** is selected, snapshots are not retained as Persistent.

6 Click **Save,** enter a name in the **Create New Set** dialog box, and click **Save**.

The name can contain alphanumeric and non-alphanumeric characters, but it cannot contain non-Latin characters. On Windows, there is no length restriction; however, a maximum of 40 characters is recommended.

Finalizing and submitting a VSS backup job

The final steps include setting additional options on the Schedule, Target Storage, and Advanced Options pages, submitting the job, and monitoring the progress by using the Job Status and View Logs pages. These pages and options are common to all NetVault Backup Plug-ins. For more information, see the *Quest NetVault Backup Administrator's Guide*.

1. Use the **Schedule**, **Target Storage**, and **Advanced Options** lists to configure any additional required options.
2. Click **Save** or **Save & Submit**, whichever is applicable.

   TIP: To run a job that you have already created and saved, select **Manage Job Definitions** in the Navigation pane, select the applicable job, and click **Run Now**.

   You can monitor progress on the **Job Status** page and view the logs on the **View Logs** page. For more information, see the *Quest NetVault Backup Administrator's Guide*.

Example of creating a full VDI backup of an AlwaysOn Availability Group

This procedure is based on the following configuration and settings:

- The AlwaysOn databases reside on the primary replica and on all secondary replicas.
- The Virtual Client uses the network name or IP address of either the AlwaysOn Availability Group Listener or the AlwaysOn Cluster.
- The database is online.
- All SQL Server services, including SQL Server Browser, are running on each node.
- NetVault Backup Server is running only on the server; it is not running on the client nodes.
- The **Log on as** option for the NetVault Process Manager is configured to use the domain administrator.
- Each NetVault Backup client node is able to access the NetVault Backup client on the other nodes.

After reviewing the preceding information, complete the steps outlined in the following topics:

- Creating a Backup Selection Set for a full VDI backup of an AlwaysOn Availability Group
- Creating a Backup Options Set for a full VDI backup of an AlwaysOn Availability Group
- Finalizing and submitting a full VDI backup job of an AlwaysOn Availability Group

Creating a Backup Selection Set for a full VDI backup of an AlwaysOn Availability Group

1. In the Navigation pane, click **Create Backup Job**.
2. In **Job Name**, specify a name for the job.
3. Next to the **Selections** list, click **Create New**.
4. In the list of plug-ins for the applicable Client, open **Plug-in *for SQL Server***, and then open the **All Instances** node to view the individual instances.
5. Do either of the following:
   - Select an entire AlwaysOn Availability Group instance for inclusion.

- Open an AlwaysOn Availability Group instance to display its individual databases, and select the databases that you want to include.

IMPORTANT: If you are running a Tail-Log backup, use SQL Server Management Studio to remove the primary and secondary databases from the group before continuing. After the databases are removed from the group, their status changes to Restoring, and they are owned by the SQL Server Instance assigned the Primary Role in the group. To complete the Tail-Log backup, select the databases under the SQL Server Instance.

6 Click **Save**, enter a name in the **Create New Set** dialog box, and click **Save**.

Creating a Backup Options Set for a full VDI backup of an AlwaysOn Availability Group

1 Next to the **Plugin Options** list, click **Create New**.

2 In the **Backup Method** section, select **Use Virtual Device Interface (VDI)**.

3 In the **Backup Type** section, select **Full Database**.

4 In the **Incomplete Backup of ALL Items Selected** section, select the applicable option.

5 In the **VDI Backup Options** section, select the applicable compression.

6 Complete the **VDI Streams/Compression Threads** and **Number of Parallel NetVault Backup Streams** options.

NOTE: In an AlwaysOn Availability Group, the plug-in uses **Number of Parallel NetVault Backup Streams** as the number of streams used by *each* NetVault Backup Client. For example, if the group has three NetVault Backup Clients, and you enter **1** for the number of streams, each uses one stream. This specification equals three streams for the overall backup job.

7 If you want the plug-in to perform checksum on the backups, select the **Enable Backup Checksum** option, and then use **Error Encountered During Checksum** to indicate what the plug-in should do if it encounters an error.

8 Click the **AlwaysOn Availability Groups** tab, and select the **Primary Only** option.

9 Click **Save**, specify a name for the set in the **Create New Set** dialog box, and click **Save**.

Finalizing and submitting a full VDI backup job of an AlwaysOn Availability Group

1 Use the **Schedule**, **Target Storage**, and **Advanced Options** lists to configure any additional required options.

2 Click **Save** or **Save & Submit**, whichever is applicable.

6

Restoring data

Restoring data: an overview

Restoring is the process of copying data from a backup and applying transaction logs to the data to roll it forward to the target **recovery point**. A backup contains sufficient transaction log records to allow rolling forward the active transactions as part of restoring each backup. Each backup also contains sufficient transaction logs to roll back any uncommitted transactions to bring the database to a consistent, usable state. The process of rolling forward uncommitted transactions, if any, and bringing the database online is known as **Recovery**.

NOTE: If your SQL Server Database Security Administrator or DBA has enabled Transparent Data Encryption (TDE) for one or more SQL Server Databases, then before restoring a TDE enabled database, SQL Server Database Security Administrator, or DBA must restore required TDE certificate on the SQL Server target instance.

Understanding the Online VDI restore process

This topic includes the following subtopics:

Reviewing the phases of the VDI restore sequence

Each SQL Server restore scenario is implemented using one or more restore operations, which is called a restore sequence. A restore sequence moves data through one or more phases of restore. The possible phases of a restore include the data copy, redo (roll forward), and undo (roll back) phases, which are detailed in the following topics.

Data copy phase of VDI restore

The first phase in any restore process is the data copy phase. The data copy phase involves copying all the data, log, and index pages from the backup media to the database files. It involves copying data from one or more of the following backup types:

- **Full Database**
- **Differential Database**

- **Copy-Only**
- **Full File and Filegroup**
- **Differential File and Filegroup**
- **Partial Database**
- **Differential Database**

Redo phase (roll forward) of VDI restore

Redo is the process of reapplying logged changes to the data in the roll-forward set to bring the data forward in time. To accomplish the redo, the SQL Server Database Engine processes Transaction Log backups as they are restored. The engine starts with the log contained in the data copy backups listed in the preceding topic and any **Incremental Transaction Log** backups that are then restored. Roll forward stops at the ***recovery point***; this point is either the end of the log or a point specified in the SQL Server restore options for **Incremental Transaction Log** backups. Plug-in *for SQL Server* supports both time-based and marked transaction-based PIT recovery.

Undo (roll back) phase of VDI restore and recovery

After the redo phase has rolled forward all the log transactions, a database typically contains changes made by transactions that are uncommitted at the recovery point. This result makes the rolled forward data transactionally inconsistent. The recovery process opens the transaction log and identifies uncommitted transactions and rolls them back to bring the database into a transactionally consistent state. This step is called the **undo phase**.

If the data is transactionally consistent at the start of the recovery process, the undo phase is skipped.

After the database is transactionally consistent, recovery brings the database online.

Understanding the available types of Online VDI restores

To perform a successful restore, you must have a full understanding of the types of restores that are available for use. This topic outlines the possible restore scenarios supported for different recovery models.

Restoring databases under Simple Recovery Model for Online VDI restore

The following restore scenarios are supported under Simple Recovery Model:

- **Complete database restore:** With a Simple Recovery Model database, the goal of a complete database restore is to restore the entire database. This type of restore typically involves restoring a **Full Database** backup or a **Full Database** backup followed by a **Differential Database** backup. The entire database is offline during a complete database restore.
- **File restore:** In a file restore for a Simple Recovery Model database, the goal is to restore one or more damaged read-only files without restoring the entire database. This type of restore involves a **Full File and Filegroup** backup.
- **Partial database restore:** A partial restore for a Simple Recovery Model database includes restoring the primary filegroup and all read/write filegroups that were included in a **Partial Database** or **Differential Partial Database** backup. Restores of the read-only filegroups with a **Full File and Filegroup** backup are required only if the read-only filegroups are damaged.

Restoring databases under Full and Bulk-Logged Recovery Models for Online VDI restore

The following restore scenarios are supported under the Full and Bulk-Logged Recovery Models:

- **Complete database restore:** Under the Full and Bulk-Logged Recovery Models, the goal is to restore the entire database to the point-of-failure. A complete restore sequence typically includes:
    1. Perform **Tail-Log** backup.
    2. Restore **Full Database** or **Full File and Filegroup** backup while selecting **With NO RECOVERY** restore option.

3 Restore **Differential Database** or **Differential File and Filegroup** backup while selecting **With NO RECOVERY** restore option.

4 Restore all subsequent **Incremental Transaction Log** backups in order while selecting the **With NO RECOVERY** restore option.

5 Restore and recover the **Tail-Log** backup by selecting the **With RECOVERY** restore option.

- **File restore:** In a file restore for a Full or Bulk-Logged Recovery Model database, the goal is to restore the entire database while leaving the option available to restore damaged files without restoring the entire database. This type of restore involves the following restore sequence:

  1 Perform **Tail-Log** backup.

  2 Restore **Full File and Filegroup** backup using the **With NO RECOVERY** restore option.

  3 Restore **Differential File and Filegroup** backup using the **With NO RECOVERY** restore option.

  4 Restore all subsequent **Incremental Transaction Log** backups in order while selecting the **With NO RECOVERY** restore option.

  5 Restore and recover the **Tail-Log** backup while selecting the **With RECOVERY** restore option.

- **Point-in-time (PIT) recovery:** You can perform a PIT recovery on the transaction logs that are restored during an **Incremental Transaction Log** restore. Plug-in *for SQL Server* supports both time-based—using hours:minutes:seconds—and marked-transaction-based PIT recovery. PIT recovery options are available when restoring **Incremental Transaction Log** and **Tail-Log** backups.

  - **Time-Based Point-in-Time Recovery:** Time-based PIT recovery is useful when the time that the data corruption occurred is known. For example, if a developer dropped a table at 6:00:00 A.M., PIT recovery can be performed with a stop time of 5:55:00 A.M. Time-based PIT recovery can only be used for unrelated databases; therefore, when multiple SQL Server databases are required to be logically consistent, marked-transaction PIT recovery should be used.

  - **Marked-Transaction-Based Point-in-Time Recovery:** SQL Server 2008 and later support the ability to insert named marks into a transaction log to allow recovery to a specific mark. This functionality is beneficial if you have two or more databases that must be logically consistent. You can implement special procedures to ensure the recoverability of these databases after a restore. Because you can only recover related databases to a marked transaction, your application must be configured to insert named marks into the transaction log before performing backups. For more information, see *Ensuring Recoverability of Related Databases and the Recovering to a Marked Transaction* in the *SQL Server Books Online*.

Restoring system databases for Online VDI restore with Plug-in *for SQL Server*

You can only restore system databases from backups that are created on the version of SQL Server that the instance is running. For example, to restore a system database on an instance that is running on SQL Server 2008, use a backup that was created after the instance was upgraded to SQL Server 2008.

Restoring the master database for Online VDI restore

The **master** database must be accessible and at least partly usable for starting a SQL Server Instance. If the **master** database becomes unusable, you can return the database to a usable state by either restoring the **master** from a current database backup or **rebuilding** the master.

You can restore the **master** database from a Full Database backup only if you are able to start the server instance. If severe damage to the **master** database prevents you from starting the SQL Server, rebuilding the **master** is the only option left. For more information, see *Rebuilding the master Database* in the *SQL Server Books Online*.

The restoration of the **master** database includes:

- Start the server instance in single-user mode.
- Restore a **Full Database** backup of the **master** database.

The SQL Server Instance is stopped automatically on completion of the data restoration process.

- Because the changes made to **master** since the last backup were lost, re-create them by performing those steps again.
- After the **master** has been restored and changes reapplied, back up the **master** immediately.

For more information, see Restoring data to an alternate SQL Server.

Restoring the model or msdb databases for Online VDI restore

Restoring the **model** or **msdb** database from a backup is required under the following circumstances:

- The **master** database has been rebuilt. In this case, restore **model** from a backup because rebuilding **master** re-creates **model**.
- The **model** or **msdb** database has been damaged, for example, due to media failure.

Restoring **model** or **msdb** is the same as performing a Complete Database restore of a user database.

Before restoring **msdb**, the **SQL Server Agent Service** in the Control Panel must be stopped. Quest also recommends that you shut down applications, because the restore procedure disconnects users when necessary.

Restoring databases involved in replication for Online VDI restore

You can restore all databases in a replication topology if recent backups are available and the proper steps are followed. The restore steps for the publication database depend on the type of replication and options used, but the steps for all other databases are independent of the type and options.

Replication supports restoring replicated databases to the same SQL Server Instance and database from which the backup was created. If you restore a replicated database from a backup to a different instance, replication settings cannot be preserved. In this case, re-create all publications and subscriptions after backups are restored.

For more information, see *Strategies for Backing Up and Restoring Snapshot and Transactional Replication* and *Strategies for Backing Up and Restoring Merge Replication* sections in the *SQL Server Books Online*.

Restoring data from Online VDI backups

A standard restore with Plug-in *for SQL Server* in SQL Server includes the steps outlined in the following topics:

- Selecting Online VDI data for restore
- Setting Online VDI restore options
- Finalizing and submitting an Online VDI restore job

Selecting Online VDI data for restore

1. In the Navigation pane, click **Create Restore Job**.
2. On the **Create Restore Job—Choose Saveset** page, select **Plug-in *for SQL Server*** from the **Plugin Type** list.
3. To filter the items displayed in the saveset table further, use the **Client**, **Date**, and **Job ID** lists.

   The table displays the saveset name (job title and saveset ID), creation date and time, and size. By default, the list is sorted alphabetically by saveset name.

The following table outlines the backup-type identifiers:

Backup type	Backup type identifier
Full Database	VDI FULL
Differential Database	VDI DIFFERENTIAL
Copy - Only	VDI Copy - ONLY
Incremental Transaction Log	VDI INCREMENTAL
Transaction Log	VDI INCREMENTAL
Tail-Log	VDI TAIL-LOG
Full File and Filegroup	VDI FILEGROUP
Differential File and Filegroup	VDI FILEGROUP DIFFERENTIAL
Partial Database	VDI PARTIAL
Differential Partial Database	VDI PARTIAL DIFFERENTIAL

4 In the saveset table, select the applicable item.

When you select a saveset, the following details are displayed in the **Saveset Information** area: Job ID, job title, server name, client name, plug-in name, saveset date and time, retirement setting, Incremental Backup or not, Archive or not, saveset size, and snapshot-based backup or not.

5 Click **Next**.

6 On the **Create Selection Set** page, select the data that you want to restore.

The database name is displayed when you open this node. For the **Full File and Filegroup**, **Differential File and Filegroup**, **Partial Database**, and **Differential Partial Database** backup types, you can open the database node and select the individual files or filegroups. For other backup types, further drill-down shows items but you ***cannot*** select them.

> **IMPORTANT:** When restoring a read-only filegroup, the filegroup must be explicitly selected in the selection tree. Though a green check mark appears for the read-only filegroup when you select the parent-level database, the filegroup is not restored. When including a read-only filegroup in a restore, select the individual items instead of selecting the parent-level database.

7 Select the applicable instance or the database for inclusion in the restore procedure.

> **IMPORTANT:** If you are restoring an AlwaysOn Availability Group, select only databases that either belong to one specific group or to one specific SQL Server Instance; you cannot select a mixture.

8 If you are restoring from a Partial Database backup and you want to restore only the modified data, use the following SQL Server command to restore only the applicable items:

```
RESTORE DATABASE <DatabaseName> <ReadWriteFilegroupOptions>
   FROM VIRTUAL_DEVICE=<VirtualDeviceName> WITH PARTIAL
   <RestoreAndRecoveryOptions>
```

The WITH PARTIAL clause identifies the specific data to restore. For example:

```
RESTORE DATABASE [testdb5] READ_WRITE_FILEGROUPS FROM
   VIRTUAL_DEVICE='SAMPLE4052' WITH PARTIAL, BLOCKSIZE=65536,
   MAXTRANSFERSIZE=1572864, REPLACE, RECOVERY
```

Setting Online VDI restore options

IMPORTANT: After you have created a restore job, you cannot update the restore options located in the **Create Restore Job—Choose Saveset** page until you select the **Modify** check box. Also, you cannot view the **Transaction Log Options** tab until you select **Modify**.

1. On the **Create Selection Set** page, click **Edit Plugin Options**.
2. Configure the following parameters—these parameters are common to all backup types:
   - **Backup Method:** This field displays the backup method used for the selected saveset.
   - **Backup Type:** This field displays the type of backup used for the selected saveset.
   - **Restore Type:** This section lets you specify the action that must be performed after the restore operation. You can select from the following options:
     - **With RECOVERY:** Select this option if no further Transaction Log or Differential Database backups are to be applied after the restore of the selected data has completed.
     - **With NO RECOVERY:** Select this option if you intend to apply a different Transaction Log or Differential Database backup after this restore has completed.
     - **With STANDBY:** Select this option when applying an **Incremental Transaction Log** backup from the primary server in a warm standby server scenario.
     - **Verify-only:** Select this option if you want to perform a verification check on a backup to determine if a backup set is complete and the entire backup is readable. The data is not restored.
   - **VDI Options:** The VDI restore method lets you take advantage of Microsoft's VDI API, which provides the maximum reliability and flexibility. You can set the following options for this method:
     - **Block Size (Bytes):** This field lets you specify the device block size, in bytes. The default value is 64KB, that is, 65536 bytes.
     - **Transfer Multiple:** The value specified in this field is used as a multiplier for the value specified in the **Block Size** field. The overall value of the two serves as the maximum allowable data transfer size. That is, the **Block Size** default of 65536 bytes multiplied by the default **Transfer Multiple** of 24, results in a maximum allowable transfer size of 1.6MB.
   - **Restore Options:** This section contains the following options:
     - **Target Instance:** This field lets you restore the database to a different SQL Server Instance. The procedure for this type of restore is outlined in Restoring a database to an alternate instance. Leave this option blank for a standard restore of the database to the same SQL Server Instance and for AlwaysOn Availability Group restores to the primary node.
     - **Undo File for Standby Restores:** For performing ***Standby Restores***, the plug-in requires a temporary Undo File. It is required only when the **Restore Type** is set to **With STANDBY**. By default, the path is blank. Specify the directory path for the temporary **UNDO.DAT** file in this field. You can set a default path for field in the Configurator. For more information, see Configuring plug-in options.
     - **Enable Restore Checksum:** Specifies that backup checksums must be verified and, if the backup lacks backup checksums, causes the restore operation to fail with a message indicating that checksums are not present. Select this check box to enable restore checksum.
     - **Error Encountered During Checksum:** If checksum is enabled, that is, the **Enable Restore Checksum** check box is selected, indicate what the plug-in should do if it encounters a checksum error. Select one of the following options from the list:
       - **Continue After Error:** This option is the default action set during the plug-in installation. With this value selected, the restore job will proceed, if the corruption permits, after returning a checksum error with the number of the page containing the invalid checksum.

- **Stop on Error:** With this value selected, the restore job reports a checksum error and stops if it encounters errors.

To change the default action for this option for all subsequent jobs, see Configuring plug-in options.

- **Restore as compressed, read-only database:** If you are using an NTFS-based system and the backup was created using NetVault Backup-based compression or no compression, select this option to restore a user database or Tail-Log as a compressed, read-only database. This option is dimmed if the backup was created using the SQL Server Backup Compression feature.

  This option is useful if you want to save space when you restore a production database to a non-production environment. Because SQL Server cannot support a read/write database that resides in a compressed folder, it is restored as read-only.

  By default, NTFS-compressed files and folders are displayed in Windows using different colors to distinguish them from standard files and folders.

  - **Restore Location:** Enter the path to a compressed folder where the data files and log files should be restored to. If the folder does not exist, NetVault Backup creates the folder as an NTFS-compressed folder. If the specified folder exists and is not an NTFS-compressed folder, the job fails and displays this message: "Database cannot be restored as compressed in this folder, specify a new folder or select an existing compressed folder."

3 If you are restoring an **Incremental Transaction Log** or **Tail-Log** backup, click the **Transaction Log Options** tab, and configure the following parameters to set the recovery point:

- **Enable Point-in-Time Restore:** Select this check box to enable PIT recovery of the log backup.
- **Point-In-Time Options:** You can perform PIT recovery with either the timestamp of the transaction or the named mark for the transaction.
  - **Date and Time:** To perform recovery up to a specific point within the log backup, select this option and set the Time—using hours:minutes:seconds—and Date in the respective fields.
  - **Marked Transaction:** To perform recovery up to a mark within the Transaction Log backup, select this option. In the **Mark** field, enter the string to identify the transaction mark. Select **Stop At** from the list to perform recovery up through this transaction—the marked transaction is included. Select **Stop Before** to recover all transactions before this mark.

Finalizing and submitting an Online VDI restore job

The final steps include setting additional options on the Schedule, Source Options, and Advanced Options pages, submitting the job, and monitoring the job's progress. You can monitor progress by using the Job Status and View Logs pages. These pages and options are common to all NetVault Backup Plug-ins. For more information, see the *Quest NetVault Backup Administrator's Guide*.

1 Click **OK** to save the settings, and then click **Next**.

2 In **Job Name**, specify a name for the job if you do not want to use the default setting.

Assign a descriptive name that lets you easily identify the job for monitoring its progress. The job name can contain alphanumeric and non-alphanumeric characters, but it cannot contain non-Latin characters. On Windows, there is no length restriction; however, a maximum of 40 characters is recommended.

3 In the **Target Client** list, select the machine on which you want to restore the data.

> **TIP:** You can also click **Choose**, and then locate and select the applicable client in the **Choose the Target Client** dialog box.

4 Use the **Schedule**, **Source Options**, and **Advanced Options** lists to configure any additional required options.

5 Click **Save** or **Save & Submit**, whichever is applicable.

You can monitor progress on the **Job Status** page and view the logs on the **View Logs** page. For more information, see the *Quest NetVault Backup Administrator's Guide*.

Restoring data from VSS backups

The Microsoft VSS restore process includes the steps outlined in the following topics:

- Selecting data for a VSS restore
- Setting VSS restore options
- Finalizing and submitting a VSS restore job

IMPORTANT: The SQL Server Instance must be stopped when restoring the **master** database.

Selecting data for a VSS restore

1 In the Navigation pane, click **Create Restore Job**.

2 On the **Create Restore Job—Choose Saveset** page, select **Plug-in *for SQL Server*** from the **Plugin Type** list.

3 To filter the items displayed in the saveset table further, use the **Client**, **Date**, and **Job ID** lists.

The table displays the saveset name (job title and saveset ID), creation date and time, and size. By default, the list is sorted alphabetically by saveset name.

The following table outlines the backup-type identifiers:

Backup type	Backup type identifier
Full Database	VSS Full
Differential Database	VSS Differential
Copy-Only	VSS Copy Only

4 In the saveset table, select the applicable item.

When you select a saveset, the following details are displayed in the **Saveset Information** area: Job ID, job title, server name, client name, plug-in name, saveset date and time, retirement setting, Incremental Backup or not, Archive or not, saveset size, and snapshot-based backup or not.

5 Click **Next**.

6 On the **Create Selection Set** page, select the applicable instance or the database for inclusion in the restore procedure.

Although you can double-click a node to open it and display the database name, and further drill-down shows items, you ***cannot*** select them.

7 If you are restoring data from a persistent snapshot, skip to Finalizing and submitting a VSS restore job.

To restore data from a persistent snapshot, no additional steps are required before you submit the job. After you submit the job, the plug-in automatically obtains the snapshot information from the backup index. If the snapshot is available, the plug-in mounts it locally and restores the selected files from the snapshot. If the snapshot is unavailable, the plug-in restores the file data from the storage media.

Setting VSS restore options

On the **Create Selection Set** page, click **Edit Plugin Options**, and configure the following parameters—these options are available for all VSS backup types:

- **Backup Method:** This field displays the backup method used for the selected saveset.
- **Backup Type:** This field displays the type of backup used for the selected saveset.
- **Restore Method:** Select one of the following options:
  - **VSS Restore:** Select this option if to complete a restore to the same machine from which the backup was created.
  - **Copy Database Files to Target Directory:** Select this option to restore database files to a specified location. You can specify a directory on the local machine where the backups were taken. You can also specify a directory on any machine that has NetVault Backup and the plug-in installed and is reachable from the NetVault Backup Server, even if the machine does not have SQL Server installed. To restore a VSS backup to a different machine, use the **Target Client** list to select the NetVault Backup Client of the corresponding target machine.

    IMPORTANT: Because the maximum path length in Windows is 260 characters, check that the combination of the filenames and target path does not exceed 260 characters; otherwise, the restore fails.

    If the backup contains FILESTREAM data, the restore process does not retain folder structure; the plug-in restores all files to the same folder. Because the folder structure of the database is not recreated, a file-activation error appears if you try to access the database. To avoid this issue, restore the backup to the original location.
- **Restore Type:** This section lets you specify the action that must be performed after the restore operation. You can select from the following options:
  - **With RECOVERY:** Select this option if you do not intend to apply additional Transaction Log or Differential Database backups after the restore of the selected data has completed.
  - **With NO RECOVERY:** Select this option if you intend to apply a different Transaction Log or Differential Database backup after this restore has completed.

Finalizing and submitting a VSS restore job

Complete the steps outlined in Finalizing and submitting an Online VDI restore job under Restoring data from Online VDI backups.

Using other restore procedures

This topic describes other restore operations that you can perform with the plug-in:

- Restoring data to a Virtual Client
- Renaming or relocating a database
- Restoring a database to an alternate instance
- Restoring data to an alternate SQL Server

Restoring data to a Virtual Client

All options available for a restore with the plug-in are also available in high-availability—SQL Server Failover Cluster and AlwaysOn Availability Group—environments. Data selection is performed the same way. The difference is that restorable backups of a Virtual Client are displayed on the **Create Restore Job—Choose Saveset** page under the name of the Virtual Client, not the specific NetVault Backup Client or node that was active during each backup. When a restore job is initiated, NetVault Backup communicates with all member Clients to determine which machine controls the cluster, and then targets this machine for the restore.

Also, you can restore a NetVault Backup Virtual Client to a non-clustered (standalone) NetVault Backup Client.

NOTE: During a restore, run the process using the node for the Virtual Client. Do *not* use one of the nodes for an individual client in the cluster to run the process. If you open or expand one of the nodes and drill down through the hierarchy, you might see a dimmed SQL Server Instance identified as **local**. While the system might use this instance in maintaining log information, do not run any processes at this level.

Important considerations restoring data to a Virtual Client for an AlwaysOn Availability Group

Use the following guidelines when you restore data that resides in an AlwaysOn Availability Group.

- You can restore data from the backup of an AlwaysOn Availability Group to the same group. You cannot restore an AlwaysOn Availability Group to an alternate SQL Server Instance, standalone deployment, or a different AlwaysOn Availability Group.
- If you complete multiple sequential restores of the primary replica and you want to omit the primary replica after the first restore, Quest recommends that you complete the **Enter the NetVault Backup Client Name on the AlwaysOn Availability Groups Primary Node** field.
- Unless you want to use SQL Server Management Studio to add the primary replica to the group, select the **Add Primary Database to the AlwaysOn Availability Group After Restore** check box.
- Ensure that the secondary replicas are always removed from the group before restoring the primary replica. You can do this manually using SQL Server Management Studio or by selecting the **Remove Primary Database from the AlwaysOn Availability Group Before Restore** check box.

  If you do not remove the secondary replicas before completing the restore, it might prevent the plug-in from removing the primary replica or the restore might complete with warnings. Also, failing to remove the secondary replicas might prevent you from adding the secondary replicas during the subsequent backup of the primary replica.

  NOTE: While restoring the backups that are created on Plug-in *for SQL Server* version 11.2 or earlier, the option **Remove Database Secondary Copies from the AlwaysOn Availability Group Before Restore** is visible on 'AlwaysOn Availability Groups' section. However, you should not select this option. This option is not displayed if the backup job is created on Plug-in *for SQL Server* version 11.4.5 or later.
- If you select **With NO RECOVERY** for the **Restore Type** of the primary replica, do *not* select the **Add Primary Database to the AlwaysOn Availability Group After Restore** check box. You must complete the recovery before you can add the database to the group.
- You cannot use the same job to restore databases from multiple AlwaysOn Availability Groups.
- You cannot use the same job to restore a database that belongs to an AlwaysOn Availability Group and a different database that does not belong to a group.
- If you use SQL Server's named instances for your AlwaysOn Availability Group nodes, the restore process requires that any database that is backed up using the Virtual Client must belong to the same named instance on *each* node of the group.
- You must create separate jobs to restore databases from an AlwaysOn Availability Group and databases that are not included in a group. Also, use the same job to restore databases from an AlwaysOn Availability Group if they reside in the same group.

Restoring data to a Virtual Client in an AlwaysOn Availability Group

The following topics list the high-level steps for restoring data in AlwaysOn Availability Group deployments and provide examples of different restores.

- High-level restore steps for restoring data to a Virtual Client in an AlwaysOn Availability Groups
- Example of restoring a database that was added to all replicas in an AlwaysOn Availability Group
- Example of restoring a secondary replica and converting it to a primary replica
- Example of restoring a VDI backup of an AlwaysOn Availability Group
- Example of restoring a database that does not exist on the primary replica

High-level restore steps for restoring data to a Virtual Client in an AlwaysOn Availability Groups

These topics describe the general steps that you must take to complete a restore.

Performing a VSS restore of an AlwaysOn Availability Group

When restoring a VSS-based backup, be aware of the following:

- VSS is a generic interface that requires objects included in the snapshot comply with certain requirements. For example, SQL Server databases that have a status of Restoring conflict with the creation of VSS backup snapshots.
- SQL Server databases participating in an AlwaysOn Availability Group cannot be included in a VSS restore snapshot, and some operations are not available for databases that are joined in a group.
- If you use the **Copy Database Files to Target Directory** option, do *not* remove the database from the group.
- As noted previously, a disadvantage of VSS backups is that the SQL Server VSS Writer does not support the ability to restore a database to an alternate instance whether that instance is on the same server and host or an alternate server and host.
- The following procedure applies if you are restoring the group to its original destination or if you are restoring the group using the Rename/Relocate option. However, if you are performing a restore with rename or relocate, or both, the original database cannot be joined in an AlwaysOn Availability Group.
    1. Use SQL Server Management Studio to remove the database from the AlwaysOn Group.
        a. On the primary replica, select **Suspend Data Movement** for the database.
        b. On all secondary replicas, remove the database from the group.
        c. On the primary replica, remove the database from the group.
    2. Restore the primary from the applicable backups.
    3. Add the database to the AlwaysOn Availability Group on the primary replica.
    4. Add the database to the AlwaysOn Availability Group on each secondary replica.

Performing a VDI restore of an AlwaysOn Availability Group

1. Remove the secondary replicas.
2. Remove the primary replica.
3. Restore the primary from the applicable backups.
4. Using the process for a standalone deployment, perform Full and Differential Transaction Log backups of the primary replica.
5. Using the process for a standalone deployment, perform Full and Differential Transaction Log restores of each secondary replica.

These restores set the secondary replicas to match the primary replica.

6 Add the database to the AlwaysOn Availability Group on the primary replica.

7 Add the database to the AlwaysOn Availability Group on each secondary replica.

Example of restoring a database that was added to all replicas in an AlwaysOn Availability Group

The following procedure describes how to restore a database that was added to the primary replica and all secondary replicas in the AlwaysOn Availability Group.

1 Select the data to restore.

For more information, see Selecting Online VDI data for restore under Restoring data from Online VDI backups or Selecting data for a VSS restore under Restoring data from VSS backups.

As described previously, you cannot use the same job to restore databases from multiple groups. Nor can you use the same job to restore a database that belongs to a group and a different database that does not.

2 On the **Create Selection Set** page, click **Edit Plugin Options**, and configure the applicable parameters depending on whether you are performing a VSS or VDI restore.

For information on the options that are common to all restore jobs, see Setting Online VDI restore options under Restoring data from Online VDI backups or Setting VSS restore options under Restoring data from VSS backups.

3 In the **AlwaysOn Availability Groups** section, select the applicable check box (es):

- **Remove Primary Database from the AlwaysOn Availability Group Before Restore**
- **Add Primary Database to the AlwaysOn Availability Group After Restore**

Where you are in the process determines which check box (es) you should select. For example, when you run the restore of the Full Backup, select both check boxes. When you run the restore of the Transaction Logs, do not select the check boxes. Also, do not select **Add Primary Database to the AlwaysOn Availability Group After Restore** if you selected the **With NO RECOVERY** option.

4 If you previously removed the database from the AlwaysOn Availability Group on the primary node, complete the following fields:

- **Restore to the following AlwaysOn Availability Group:** Enter the name of the group.
- **Enter the NetVault Backup Client Name on the AlwaysOn Availability Groups Primary Node (optional):** Enter the name of the client that is running on the primary node. If you enter a name for a different node, the restore fails. If you are unsure which client is running on the primary node, use SQL Server Management Studio to determine which node is the primary node in the group.

> **IMPORTANT:** Although the second field is labeled optional, it is required in this scenario.
>
> If you enter the correct information and the database is already part of the group, but you do not select the **Remove Primary Database from the AlwaysOn Availability Group Before Restore** check box, the restore fails.
>
> If the database is not part of the group, you specify an incorrect name, and the plug-in is unable to check that the client resides on the primary replica, the plug-in restores the database. However, you cannot add the database to the group.
>
> If a failover to a different node occurs after the restore job is run and you have entered the primary client and group names, update the information stored in **Restore Options** to reflect the name of the new client. Otherwise, subsequent runs of the same restore job fail.

5 In the **Instance Name, If Restoring to an AlwaysOn Named Instance** field, enter the name of the instance.

> **TIP:** Only enter the name of the instance. If you enter the fully qualified name, *<HostName>**<InstanceName>*, the plug-in ignores the hostname.

6 Complete the steps outlined in Finalizing and submitting an Online VDI restore job under Restoring data from Online VDI backups.

7 Perform a Full Database backup from the primary replica.

8 Perform a restore from *each* secondary replica that was backed up by the Full Database backup performed in Step 7.

Complete a separate restore operation for each secondary replica that makes up the primary replica.

9 Use SQL Server Management Studio to add the secondary replicas to the AlwaysOn Availability Group.

Example of restoring a secondary replica and converting it to a primary replica

The following procedure describes how to restore a secondary replica to a backup taken from the Virtual Client and then convert the secondary replica to function as the primary replica.

1 On the secondary replica, remove the database from the AlwaysOn Availability Group.

2 Select the data to restore.

For more information, see Selecting Online VDI data for restore under Restoring data from Online VDI backups or Selecting data for a VSS restore under Restoring data from VSS backups.

As described previously, you cannot use the same job to restore databases from multiple groups. Nor can you use the same job to restore a database that belongs to a group and a different database that does not.

3 On the **Create Selection Set** page, click **Edit Plugin Options**, and configure the applicable parameters depending on whether you are performing a VSS or VDI restore.

For information on the options that are common to all restore jobs, see Setting Online VDI restore options under Restoring data from Online VDI backups or Setting VSS restore options under Restoring data from VSS backups.

4 In the **Target Instance** field in the **Restore Options** section, enter the name of the client running on the secondary replica.

5 In the **AlwaysOn Availability Groups** section, select the applicable check boxes:

- **Remove Primary Database from the AlwaysOn Availability Group Before Restore**
- **Add Primary Database to the AlwaysOn Availability Group After Restore**

Where you are in the process determines which check boxes you should select. For example, when you run the restore of the Full Backup, select all the check boxes. When you run the restore of the Transaction Logs, do not select the check boxes. Also, do not select **Add Primary Database to the AlwaysOn Availability Group After Restore** if you selected the **With NO RECOVERY** option.

6 In **Restore to the following AlwaysOn Availability Group**, enter the name of the group.

7 In the **Instance Name, If Restoring to an AlwaysOn Named Instance** field, enter the name of the instance.

TIP: Only enter the name of the instance. If you enter the fully qualified name, <*HostName*>\<*InstanceName*>, the plug-in ignores the hostname.

8 Complete the steps outlined in Finalizing and submitting an Online VDI restore job under Restoring data from Online VDI backups.

9 Perform a failover to the secondary replica.

10 Use SQL Server Management Studio to identify the secondary replica as the primary replica for all databases assigned to the AlwaysOn Availability Group.

11 Use SQL Server Management Studio to add the database back to the group on the new primary replica.

IMPORTANT: To complete this step, you might need to remove the database from the remaining secondary replicas and postpone adding the database to the new primary replica until the process is finished.

12 Perform Full Database and Transaction Log backups of the new primary replica.

13 Perform a restore of *each* secondary replica using the Full and Transaction Log backups completed in the previous step.

   For each secondary replica, remove the database on that node from the AlwaysOn Availability Group, and complete the Full and Transaction Log restores.

14 Use SQL Server Management Studio to add the secondary replicas to the AlwaysOn Availability Group.

Example of restoring a VDI backup of an AlwaysOn Availability Group

The following procedure describes how to restore a VDI backup. This procedure is based on the example described in Example of creating a full VDI backup of an AlwaysOn Availability Group. The following procedure also requires that you restore Transaction Log backups, a process that was not included in the previous backup example.

1 In the Navigation pane, click **Create Restore Job**.

2 In the saveset table on the **Create Restore Job—Choose Saveset** page, select the Full Backup of the primary replica that you are restoring, and click **Next**.

3 On the **Create Selection Set** page, select the AlwaysOn Availability Group that you want to restore.

   Include all databases included in the saveset.

4 Click **Edit Plugin Options**.

5 In the **Restore Type** section, select **With NO RECOVERY**.

6 In the **AlwaysOn Availability Groups** sections, select the **Remove Primary Database from the AlwaysOn Availability Group Before Restore** check box.

   Do not select the **Add Primary Database to the AlwaysOn Availability Group After Restore** because you selected the **With NO RECOVERY** option.

7 Click **OK** to save the settings, and then click **Next**.

8 In **Job Name**, specify a name for the job if you do not want to use the default setting.

9 In the **Target Client** list, select the same Virtual Client that was the target of the backup.

10 Click **Save & Submit**, and use the **Job Status** page to monitor progress.

   You might see a “Waiting for secondaries” status for up to 20 minutes. Do not proceed until the Run Status is “Restore Completed.”

11 In the Navigation pane, click **Create Restore Job**.

12 In the saveset table on the **Create Restore Job—Choose Saveset** page, select the Incremental Transaction Log backup of the primary replica that you are restoring, and click **Next**.

13 On the **Create Selection Set** page, select the same AlwaysOn Availability Group and databases that you selected during the restore of the Full Backup.

14 Click **Edit Plugin Options**.

15 In the **Restore Type** section, select **With RECOVERY**.

16 In the **AlwaysOn Availability Groups** section, select the **Add Primary Database to the AlwaysOn Availability Group After Restore** check box.

   Do not select the other check boxes.

17 In the **Restore to the following AlwaysOn Availability Group** field, enter the name of the group.

18 In the **Enter the NetVault Backup Client Name on the AlwaysOn Availability Groups Primary Node** field, enter the name of the client on the primary node.

   Check that you entered the NetVault Backup Virtual Client name; do not enter a different name, such as the host name or SQL Server Instance.

19 If you are restoring the backup to a named SQL Server Instance, not the default instance, enter the name of the instance in the **Instance Name, If Restoring to an AlwaysOn Named Instance** field.

TIP: Only enter the name of the instance. If you enter the fully qualified name, <*HostName*>\<*InstanceName*>, the plug-in ignores the hostname.

20 Click **OK** to save the settings, and then click **Next**.

21 In **Job Name**, specify a name for the job if you do not want to use the default setting.

22 In the **Target Client** list, select the same Virtual Client that was the target of the backup.

23 Click **Save & Submit**, and use the **Job Status** page to monitor progress.

Do not proceed until the Run Status is "Restore Completed."

After the restore of the primary replica is finished, create a backup of the primary replica. After the backup is finished, use the new backup to restore each secondary replica.

24 In the Navigation pane, click **Create Backup Job**.

25 In **Job Name**, specify a name for the job.

26 Next to the **Selections** list, click **Create New**.

27 In the selection tree, open the first *secondary* replica that you are restoring, and then open the **All Instances** node.

The databases that were part of the AlwaysOn Availability Group showing "(Restoring)" after their names.

28 Select the **All Instances** node, and select **Configure** from the context menu.

29 Enter the security information, enter the applicable login information, such as **(local)**, for the **Instance Name**, and click **OK**.

30 Repeat Step 27 through Step 29 for each *secondary* replica.

31 In the selection tree for the *primary* replica, select that databases that you are backing up.

These databases are the same ones identified as "(Restoring)" on the secondary replica.

32 Run a VDI-based, **Full Database** backup.

33 Using the same **Selections** set that you used for the Full Database backup, run an **Incremental Transaction Log** backup.

34 In the Navigation pane, click **Create Restore Job**.

35 In the saveset table on the **Create Restore Job—Choose Saveset** page, select the Full Backup of the primary replica that you completed Step 32, and click **Next**.

36 On the **Create Selection Set** page, select the databases that you want to restore, and click **Edit Plugin Options**.

37 In the **Restore Type** section, select **With NO RECOVERY**, click **OK**, and then click **Next**.

38 In **Job Name**, specify a name for the job if you do not want to use the default setting.

39 In the **Target Client** list, select the Virtual Client name of the *secondary* replica.

40 Click **Save & Submit**, and use the **Job Status** page to monitor progress.

41 Repeat Step 34 through Step 40 to restore the Incremental Transaction Log backup from the primary replica to the secondary replica.

42 From the secondary replica, use SQL Server Management Studio to add each applicable database to the AlwaysOn Availability Group.

Perform this step locally on the applicable secondary node.

43 Repeat Step 34 through Step 42 for each secondary replica that is part of the group.

Example of restoring a database that does not exist on the primary replica

The following procedure describes how to restore a database to an AlwaysOn Availability Group if the database was removed from the primary replica before the restore.

1. Select the data to restore.

   For more information, see Selecting Online VDI data for restore under Restoring data from Online VDI backups or Selecting data for a VSS restore under Restoring data from VSS backups.

   As described previously, you cannot use the same job to restore databases from multiple groups. Nor can you use the same job to restore a database that belongs to a group and a different database that does not.

2. On the **Create Selection Set** page, click **Edit Plugin Options**, and configure the applicable parameters depending on whether you are performing a VSS or VDI restore.

   For information on the options that are common to all restore jobs, see Setting Online VDI restore options under Restoring data from Online VDI backups or Setting VSS restore options under Restoring data from VSS backups.

3. In the **AlwaysOn Availability Groups** section, *clear* the check box **Remove Primary Database from the AlwaysOn Availability Group Before Restore**.

4. Select or clear the **Add Primary Database to the AlwaysOn Availability Group After Restore** check box, whichever is applicable.

5. Complete the following fields (required):

   - **Restore to the following AlwaysOn Availability Group:** Enter the name of the group.
   - **Enter the NetVault Backup Client Name on the AlwaysOn Availability Groups Primary Node (optional):** Enter the name of the client that is running on the primary node. If you enter a name for a different node, the restore fails. If you are unsure which client is running on the primary node, use SQL Server Management Studio to determine which node is the primary node in the group.

6. If you are restoring the backup to a named SQL Server Instance, not the default instance, enter the name of the instance in the **Instance Name, If Restoring to an AlwaysOn Named Instance** field.

   TIP: Only enter the name of the instance. If you enter the fully qualified name, *<HostName>**<InstanceName>*, the plug-in ignores the hostname.

7. Complete the steps outlined in Finalizing and submitting an Online VDI restore job under Restoring data from Online VDI backups.

8. Perform a Full Database backup from the primary replica.

9. Perform a restore from *each* secondary replica that was backed up by the Full Database backup performed in Step 7.

   Complete a separate restore operation for each secondary replica that makes up the primary replica.

10. Use SQL Server Management Studio to add the secondary replicas to the AlwaysOn Availability Group.

Renaming or relocating a database

The plug-in lets you restore a database to a different name during restore. The renaming of a database can be useful if you do not want to overwrite the existing version and want to create a copy of the database. You can also relocate the database to a different directory while restoring it with the plug-in.

IMPORTANT: To make a copy of a database using the Full File and Filegroup Backup method, available only for VDI, it must be restored to a different SQL Server Instance.

When renaming a database during a VSS-based restore, the database files must also be relocated to an alternate directory. If the database files are not relocated to an alternate directory, the restore fails. This failure is because VSS does not let you rename the database without moving the data and log files to a different directory. When relocating data and log files for VSS-based restores, you can relocate the data files to one directory and the log files to a separate directory if both directories are different than the original.

1 In the Navigation pane, click **Create Restore Job**, select **Plug-in *for SQL Server*** from the **Plugin Type** list, select the applicable saveset, and click **Next**.

   Select the database to restore from the backup saveset by following the instructions outlined in Restoring data from Online VDI backups.

2 With the database selected on the **Create Selection Set** page, select **Rename** from the context menu.

3 In the **Rename/Relocate** dialog box, enter the applicable information:

   - **Rename:** Enter the new name.

     IMPORTANT: When renaming a database during a VSS-based restore, the database files must also be relocated to an alternate directory. If the database files are not relocated to an alternate directory, the restore fails. This failure is because VSS does not let you rename the database without moving the data and log files to a different directory. When relocating data and log files for VSS-based restores, you can relocate the data files to one directory and the log files to a separate directory if both directories are different from the original.

   - **Relocate:** Enter the new path.

     The plug-in lets you relocate the log files to a directory other than the data files directory. To restore to separate directories, type in the paths to the directories where the data files and log files are to be restored. Enter the data file directory first, and then enter the path to the log file directory. Separate the two directory paths using a comma—no space must exist before or after the comma. The directory specified in this field must exist at job run-time.

4 To apply your changes, click **OK**.

   NOTE: In the WebUI, the database name is accompanied by the rename and relocate information in parentheses. The database files are renamed using a concatenation of the new name and the original name, that is, <NewName>_<OriginalName>.

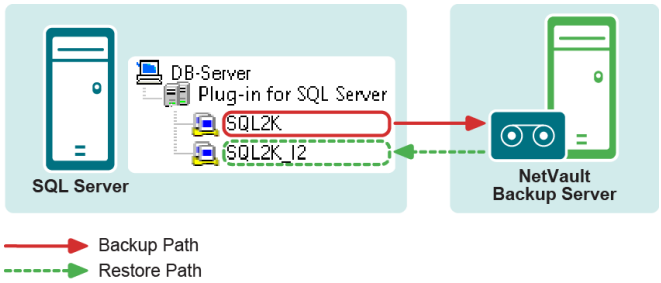
5 Continue with the restore procedure as explained in Restoring data from Online VDI backups.

Restoring a database to an alternate instance

With the plug-in's VDI-based backup method, you can restore a database to an alternate instance of SQL Server running on the database server. The procedure for performing this type of a restore is outlined in the following topic. This process does *not* support the use of AlwaysOn Availability Groups.

IMPORTANT: If you intend to restore a database to an alternate instance that resides on the same server and the original database still exists, use the **Rename/Relocate** option described previously in Renaming or relocating a database. Use this option to rename or move the datafiles to ensure that they do not overwrite or conflict with the same files in the original database.

Figure 4. Restoring to alternate instance



1 Before initiating this type of restore, check that the following prerequisites are met:

   - **Same version of SQL Server:** Both the source instance from which the data was backed up and the target instance to which the data is to be restored must be running the same version of SQL Server.

- **Instance configured for access:** The instance to which the database is to be restored must be accessible to the plug-in. To set up the instance for access, see Configuring authentication details.

2 In the Navigation pane, click **Create Restore Job**, select **Plug-in *for SQL Server*** from the **Plugin Type** list, select the applicable saveset, and click **Next**.

   Select the database to restore from the backup saveset by following the instructions outlined in Restoring data from Online VDI backups.

3 On the **Create Selection Set** page, click **Edit Plugin Options**, and configure the restore options by following the instructions outlined in Restoring data from Online VDI backups.

4 In addition, set the **Target Instance** parameter.

   To do so, specify the name of the instance to which the selected database is to be restored. You can either specify the fully qualified name, that is, *<HostName>**<InstanceName>*—for example, WIN2K3\SQLINSTANCE2 or the instance name, for example, SQLINSTANCE2, in this field.

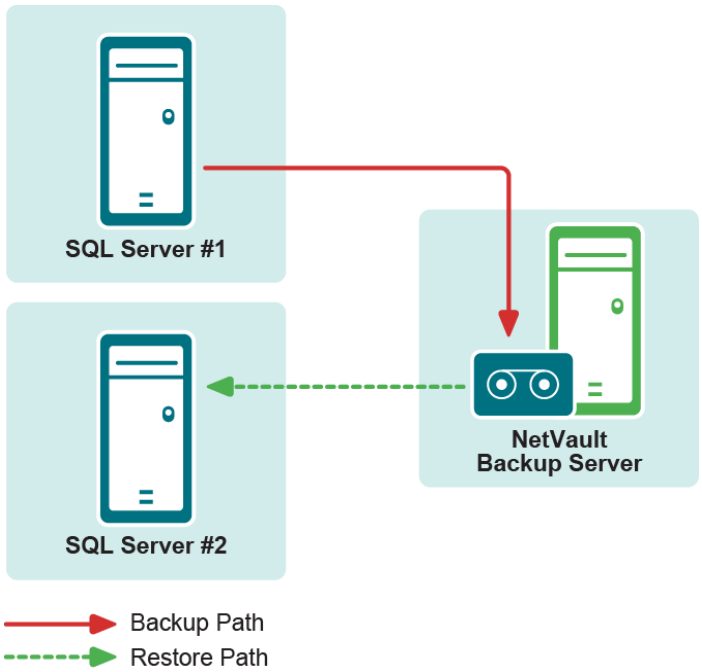
5 Complete the procedure by following the instructions outlined in Restoring data from Online VDI backups.

> **IMPORTANT:** You can also rename or relocate a database while performing this type of restore for VDI-based backups. For more information, see Renaming or relocating a database.

Restoring data to an alternate SQL Server

Similar to the plug-in's ability to restore databases to a different SQL Server Instance on the same database server, you can target a ***different SQL Server*** during the restore process. This process does *not* support the use of AlwaysOn Availability Groups.

Figure 5. Restoring to alternate SQL Server



This functionality is useful during ***disaster recovery*** operations. You can restore a database to a different SQL Server by using the steps provided in the following topics. You can also use this process to restore a NetVault Backup Virtual Client to a standalone NetVault Backup Client.

IMPORTANT: Microsoft's VSS backup method does not support restoring a VSS-based backup to an alternate instance; therefore, restoring VSS-based backups to an alternate instance with Plug-in *for SQL Server* is not supported.

In a disaster recovery scenario, use this procedure to restore the **master** and the **msdb** database backups. After restoring the **master** database, restart the server. Restore and recover each individual database.

1 Before initiating this type of restore, check that the following prerequisites are met on the new restore target:

- **Same version of SQL Server installed:** The SQL Server version must be the same as the version running on the existing database server.
- **NetVault Backup software and the Plug-in *for SQL Server* installed:** The same version of NetVault Backup software and the plug-in must be installed and configured on the new restore target.
- **Client added to the NetVault Backup Server:** The target machine must be added to the NetVault Backup Server by using the **Manage Clients** page of the NetVault Backup WebUI.
- **Instance of SQL Server must exist on the new restore target:** An instance of SQL Server must exist on the ***new restore target***. This instance serves as the target of the relocation restore. Not only must this instance be properly set up and configured in SQL Server, it must also be accessible to the plug-in on the new restore target. To set up the instance for access, see Configuring authentication details.

2 Restore the **master** and the **msdb** database backups.

IMPORTANT: When you perform a restore to an alternate server, you cannot modify the name or destination of the restored **master** database. The **master** database is restored to the location of the current **master** database in the target for the restore.

3 After restoring the **master** database, restart the server.

4 Restore and recover each individual database.

a In the Navigation pane, click **Create Restore Job**, select **Plug-in *for SQL Server*** from the **Plugin Type** list, select the applicable saveset, and click **Next**.

Select the database to be restored from the backup saveset by following the instructions outlined in Restoring data from Online VDI backups.

b On the **Create Selection Set** page, click **Edit Plugin Options**, and configure the restore options by following the instructions outlined in Restoring data from Online VDI backups.

c In addition, set the **Target Instance** parameter.

To do so, specify the name of the instance to which the selected database is to be restored on the new database server. Enter only the instance name in this field; the restore fails if the server name is included.

NOTE: If you are restoring the default SQL Server Instance to the alternate server, you can enter the name of the destination server, for example, **MSSQLSERVER**, for the **Target Instance**.

d Click **OK** to save the settings, and then click **Next**.

e In **Job Name**, specify a name for the job if you do not want to use the default setting.

Assign a descriptive name that lets you easily identify the job when monitoring its progress. The job name can contain alphanumeric and non-alphanumeric characters, but it cannot contain non-Latin characters. On Windows, there is no length restriction; however, a maximum of 40 characters is recommended.

f In the **Target Client** list, select the new restore target.

g Use the **Schedule**, **Source Options**, and **Advanced Options** lists to configure any additional required options.

h Click **Save** or **Save & Submit**, whichever is applicable.

You can monitor progress on the **Job Status** page and view the logs on the **View Logs** page. For more information, see the *Quest NetVault Backup Administrator's Guide*.

A

Troubleshooting

This topic describes some common errors and their solutions. If an error occurs that is not described in this table, obtain the SQL Server error number from the NetVault Backup Logs, and then see the relevant SQL Server documentation for the resolution.

Table 1. Troubleshooting

Symptom	Reason/symptom	Solution
Transaction Log backup failed	Transaction Log backups are not allowed for Simple Recovery Model databases.	For taking Transaction Log backups, the recovery model for the database must be set to Full Recovery Model. Or clear the Simple Recovery Model databases.
Backup failed	The File and Filegroup backup fails and the following error message appears in the logs: "The primary filegroup cannot be backed up as a file backup because the database is using the SIMPLE recovery model. Consider taking a partial backup by specifying READ_WRITE_FILEGROUPS."	Either modify the Selection Set to include only secondary filegroups for the Simple Recovery Model databases or select the Partial Database or Differential Partial Database backup type.
Backup failed	Login failed for user <*userName*>.	Verify that the password for the plug-in has been entered correctly.
Backup Completed with warnings	After this status is revealed for backup, check the NetVault Backup Log entries for the job to see if they display one or both of the following messages: • "Failed to add backup record" • "Failed to write index of backup to the database" These messages indicate that the selected data was backed up, but the job's index information was not properly added by NetVault Backup to its database. Without this index information, the data cannot be properly restored.	**Method 1:** Open the **Manage Devices** page, select the backup media, and click **Scan**. NetVault Backup stores index information for backup jobs in two locations: in the NetVault Database and on the media targeted by the backup. Performing this scan adds the index information to the NetVault Database. To verify that the information was added, open the **Manage Job Definitions** page, and locate the specific job. If you can run the job now, the scan process has corrected the problem. **Method 2:** If the scan has failed, run the backup job again.
VSS Restore fails from a NetVault Backup Server installed on Linux or UNIX.	By default, the plug-in assumes that the media format for backup tapes is MTF, which is typically used for Windows. However, Linux and UNIX use CPIO.	Configure the plug-in to set the Media Format to use CPIO during VSS Restores. For more information, see Configuring plug-in options.

Table 1. Troubleshooting

Symptom	Reason/symptom	Solution
AlwaysOn Availability Group not displayed under the Virtual Client	If a group is not listed under a Virtual Client, it might indicate that the services for the SQL Server Instances are not running. It might also indicate that valid login credentials were not set up for the plug-in at the All Instances level.	• Use **SQL Server Configuration Manager** to verify that services are running. • Verify that you set up valid credentials for the **All Instances** node located under the applicable Virtual Client in the selection tree. For more information, see Configuring the plug-in.
Nodes take a long time to open	To populate the next level of the node, the plug-in queries SQL Server for instance and cluster information. If the plug-in queries or logs in to an instance whose Windows service is not running, a lengthy timeout imposed by SQL Server might occur. NetVault Backup uses a five-minute timeout. If the NetVault Backup timeout is exceeded, the plug-in displays an error message and the node is not opened. Address the issues that are causing a delay in the login and query processes.	• Use **SQL Server Configuration Manager** to verify that services are running. • Enter default credentials for plug-in logins to SQL Server. For more information, see Entering logon credentials for all instances on a client. • If different credentials are required for some of the instances, enter the login credentials for those instances using the applicable instance or AlwaysOn Availability Group node in selection tree. For more information, see Entering default logon credentials for a specific SQL Server Instance. • If you must open the All Instances node to configure login credentials for specific instances or AlwaysOn Availability Groups, but the NetVault Backup timeout setting is preventing the All Instances node from opening, use the **Physical Client Long Timeout** option to change the timeout value for the node. For more information, see the topic on configuring the timeout period in the *Quest NetVault Backup Administrator's Guide*.
VSS backup and restore jobs might fail due to concurrent backup jobs running on the OS from NetVault Backup.	If two backup/restore jobs utilizing snapshots are scheduled to run at the same time, one of the jobs might fail with a VSS_E_SNAPSHOT_SET_IN_PROGRESS error in the log. This message indicates that VSS will not support taking two different snapshot sets at the same time.	It is recommended not to run multiple jobs at the same time on a given client. Reschedule the job OR rerun the job after 30 minutes if it is a one time job.
VSS backup and restore jobs in SQL Server might fail with an error in the log.	Backup and restore jobs for SQL Server with VSS option might fail with on of the following error in the log. VSS_E_WRITERERROR_RETRYABLE VSS_E_SNAPSHOT_SET_IN_PROGRESS VSS_E_FREEZE_TIMEOUT	Rerun/ retry the failed backup/ restore job after 30 minutes.

Table 1. Troubleshooting

Symptom	Reason/symptom	Solution
Upgrading the plug-in fails.	If a Plug-in *for SQL Server* process is running, upgrading the plug-in might fail, and an error message might be displayed during the upgrade: Failed to install software. A key could not be installed.	In the NetVault Backup Client where SQL Server is running, check if one or more Plug-in *for SQL Server* processes, named `nvsqlserver`, are running. End the running Plug-in *for SQL Server*, nvsqlserver, processes. You can use Windows Process Explorer utility, or NetVault Backup `nvpview` utility, to view and end the processes. Alternatively, restart (stop, then start) the NetVault Backup services in the NetVault Backup Client where SQL Server is running.
VDI backup job shows the following message in the NetVault Backup binary logs: `ODBC error: Could not insert a backup or restore history/detail record in the msdb database.`	The backup includes one or more databases with a name exceeding 117 characters in length. The ODBC error message is shown for each database with a name exceeding 117 characters in length.	If performing backups using the VDI backup method, limit the names of the databases in your environment to 117 characters in length.

About us

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.

Third-party contributions

This product contains the following third-party components. For third-party license information, go to https://www.quest.com/legal/license-agreements.aspx. Source code for components marked with an asterisk (*) is available at https://opensource.quest.com.

Table 2. List of third-party contributions

Component	License or acknowledgment
zlib 1.2.5	copyright © 1995-2010 Jean-loup Gailly and Mark Adler.