Veritas NetBackup™ for Microsoft SQL Server Administrator's Guide

for Windows

Release 8.1.1

VERITAS™

Veritas NetBackup™ for Microsoft SQL Server Administrator's Guide

Last updated: 2018-04-10

Document version:NetBackup 8.1.1

Legal Notice

Veritas Technologies LLC

500 E Middlefield Road
Mountain View, CA 94043

http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

NB.docs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1

Introducing NetBackup for SQL Server

This chapter includes the following topics:

Overview of NetBackup for SQL Server

NetBackup for SQL Server extends the capabilities of NetBackup for Windows to include backups and restores of SQL Server databases. These capabilities are provided for a Windows client using either a UNIX or Windows NetBackup master server. NetBackup for SQL Server includes a client-based graphical user interface (GUI) program to perform various activities on SQL Server.

NetBackup offers the following types of SQL Server backup policies:

- SQL Server Intelligent Policies. A single policy protects multiple SQL Server instances that are spread over multiple clients. You select instances for a policy from a list of instances that are automatically discovered in the NetBackup environment.
- Legacy policies, using clients and batch files. These policies include a list of SQL database clients and a batch file that contains SQL backup commands to run when the backup is scheduled. The user creates this batch file manually or through the NetBackup MS SQL Client interface, which saves the options the user selects to the batch file.

NetBackup for SQL Server includes the NetBackup MS SQL Client to perform various activities on SQL Server, as follows:

- Restores of databases and database components, which include transaction logs, differentials, files, and filegroups.
- Configuration of restore options.
- Monitoring of NetBackup for SQL Server restore operations.
- (SQL Server legacy policies) Backups of databases and database components and configuration of backup options.

In this guide, Microsoft SQL Server is referred to as SQL Server. NetBackup for Microsoft SQL Server is referred to as NetBackup for SQL Server.

Features of NetBackup for SQL Server

Table 1-1 describes the features for NetBackup for SQL Server.

Table 1-1 NetBackup for SQL Server features

Feature	Description
NetBackup integration	Full integration with the NetBackup master server and media manager. Job monitoring from the server and the NetBackup MS SQL Client interface.
SQL Server Intelligent Policy	Intelligent Policy offers the following benefits: ■ Create a single policy to protect multiple SQL Server instances or the databases in an instance. These instances can be spread over multiple clients. ■ Include a full, differential, and transaction log backup in the same policy. ■ Schedule frequent backups of transaction logs. ■ You are not required to know SQL Server commands or to write and use batch files. Instead, this feature automatically generates the batch files at run-time.
SQL Server instance management	NetBackup automatically discovers SQL Server instances in the environment. The user can view and register these SQL Server instances, which are used to build a SQL Server Intelligent Policy. The user can also use instance groups to organize instances and, optionally, automatically register instances.
Authentication and credentials	SQL Server Intelligent Policy supports the following: ■ Windows authentication and Windows Active Directory authentication. ■ With the proper configuration, you do not have to run the NetBackup Service Account as a privileged SQL Server user on the client. ■ The SQL Server DBA can manage the SQL Server credentials and instance registration independently from the NetBackup administrator, with the `nbsqladm` command.

Table 1-1 NetBackup for SQL Server features *(continued)*

Feature	Description
Backup and restore features	NetBackup supports the backup of databases, files, filegroups, transaction logs. In addition, you can perform copy-only backups and backups of only read-write filegroups. An administrator that uses the NetBackup MS SQL Server Client can browse backups and select the ones to be restored.
Automated backups	Administrators can set up schedules for automatic, unattended backups for instances on local or remote hosts across the network. These backups can be full, differential, or transaction log backups and are managed entirely by the NetBackup server from a central location. The administrator can also perform manual backups.
Snapshot backups and restores	NetBackup can perform backups of SQL Server with snapshot methodology. Also available are off-host backups, Instant Recovery, and backups with a hardware provider.
Stream-based backups and restores	Stream-based backup and restore of SQL Server objects to tape or disk with SQL Server's high-speed virtual device interface.
Redirected restores	Restore SQL Server objects to different locations.
Legacy SQL Server policies	Support for the legacy backup policies that use batch files and a list of clients.
Performance tuning	Performance tuning with policy configuration, including the following options: backup stripes, transfer size, buffer usage, and the option to skip any databases that are unavailable.
Support for high availability (HA) environments	SQL Server Intelligent Policies can protect the following HA environments: ■ SQL Server clusters ■ Log-shipping Note that the same caveats exist for Intelligent Policies as for legacy SQL Server policies. Legacy SQL Server policies can protect the following environments: ■ SQL Server clusters ■ SQL Server AlwaysOn Availability Groups™ (AGs) ■ Database mirroring ■ Log-shipping For information on the HA solutions that SQL Server supports, refer to your SQL Server documentation.

Table 1-1 NetBackup for SQL Server features *(continued)*

Feature	Description
Compression	Compression increases backup performance over the network and reduces the size of the backup image that is stored on the disk or tape. The user can select NetBackup compression or SQL Server compression. Both options should not be enabled for the same policy.
NetBackup encryption	When the Encryption attribute is enabled, NetBackup encrypts the backup for the instances or clients that are listed in the policy.
Multistreaming	Ability to use multiple stripes during a backup.
Support for VMware backups that protect SQL Server	Support for application-consistent backups of VMware computers using the VMware intelligent policy. The VMware intelligent policy includes three features that NetBackup for SQL Server supports: VMware snapshots, Replication Director (RD) snapshots, and Accelerator. Only full backups are supported on these three variations of the VMware intelligent policy. Hyper-V is not supported at this time. See the following documents for more information on VMware intelligent policy, RD, and Accelerator. NetBackup for VMware Administrator's Guide NetBackup Replication Director Solutions Guide NetBackup Administrator's Guide, Volume I

NetBackup for SQL Server terminology

Table 1-2 shows the important terms that might be new to a SQL Server database administrator or a NetBackup administrator.

Table 1-2 NetBackup for SQL Server terminology

Term	Definition
batch file	The script that is used to back up or to restore SQL Server objects. The database agent performs all operations through a batch file. Batch files are typically stored in the *install_path*\dbext\mssql\ directory. For operations executed immediately from the NetBackup Microsoft SQL Client, a temporary batch file is placed in the following directory: \Veritas\Netbackup\dbext\mssql\temp directory
full backup	A complete backup of the database that contains all of the data files and the log file. (Note that a full backup does not truncate the transaction log.)

Table 1-2 NetBackup for SQL Server terminology *(continued)*

Term	Definition
differential backup	A backup of the changed blocks since the last full backup.
transaction log	An ongoing record of updates that were made to a database.
transaction log backup	A backup of the inactive portion of the transaction log. Typically, this portion of the transaction log is truncated after it has been backed up successfully.
restore	To copy data back to a SQL Server object (see "recovery").
recovery	To bring a database online as a result of a restore.
SQL host	The host machine on which SQL Server resides. It may also refer to the virtual name of a cluster that supports a SQL Server installation.
SQL instance	A SQL Server installation. If an instance is not specified, it is considered the default SQL instance for the SQL host.
source client	A NetBackup term that identifies a host machine. The source client is commonly the network name of the host. It can also be an IP address or a cluster name, depending on how it is identified in the client configuration.
backup stripes	A data stream that is used for a backup or a restore of SQL Server objects. The user specifies the number of stripes for the backup. NetBackup performs a separate job for each stripe that is specified.
multiplex	When more than one backup stripe is written simultaneously to the same tape.
multistream	The generic method in which NetBackup manages a backup or restore that includes multiple backup stripes. Multiplexing is an example of multistreaming. NetBackup can also perform a multistreamed backup by writing individual streams to individual drives.
ODBC	An open interface protocol that NetBackup for SQL Server uses to interact with SQL Server.
VDI	Virtual device interface. A proprietary interface that SQL Server provides for backup and for restore. The interface is used both for snapshot and for streamed operations. A VDI connection is managed as a COM object.

Help for the NetBackup MS SQL Client

An online Help file for the **NetBackup MS SQL Client** interface is located in the following directory:

install_path\Veritas\Help\nbmssql.chm

Chapter 2

Installing NetBackup for SQL Server

This chapter includes the following topics:

Planning the installation of NetBackup for SQL Server

Table 2-1 shows the major installation steps that are needed to run NetBackup for SQL Server. Each step contains one or more links to pertinent procedures and concepts.

Table 2-1 Installation steps for NetBackup for SQL Server

Step	Action	Description
Step 1	Verify the operating system and platform compatibility.	See "Verifying the operating system and platform compatibility" on page 20. See the NetBackup Compatibility Lists.
Step 3	Verify the NetBackup server and client requirements for NetBackup for SQL.	See "NetBackup server and client requirements" on page 21.

Table 2-1 Installation steps for NetBackup for SQL Server *(continued)*

Step	Action	Description
Step 4	Install the NetBackup client software on the computers that have the databases that you want to back up.	Note the following: ■ In a VMware environment, install the NetBackup client software on the virtual machines that have SQL Server running. ■ For SQL Server availability groups (AGs), install the NetBackup client on each node in the AG. ■ In a SQL Server cluster environment, install the NetBackup client on each node in the cluster. ■ If you have multiple NICs, install the NetBackup client using the private interface name. ■ If the SQL Server client is on a different host than the master server or media server, then install the NetBackup client on that host.
Step 5	Verify that the SQL Server software is installed and operational on the NetBackup client(s).	
Step 6	If you installed NetBackup in a cluster, review the requirements for that environment.	See "Requirements for using NetBackup for SQL Server in a NetBackup cluster" on page 22.
Step 7	Verify that master server has a valid license for NetBackup for SQL Server and any NetBackup options or add-ons that you want to use.	See "About the license for NetBackup for SQL Server" on page 22.

Verifying the operating system and platform compatibility

Verify that the NetBackup for SQL Server agent is supported on your operating system or platform.

To verify operating system and compatibility

1. Go to the following webpage:

   http://www.netbackup.com/compatibility

2. In the list of documents, click on the following document:

   Application/Database Agent Compatibility List

3. For information on support for Snapshot Client, see the following document:

   Snapshot Client Compatibility List

4. For information on support for VMware, see the following document:

   Statement of Support for NetBackup in a Virtual Environment (Virtualization Technologies)

NetBackup server and client requirements

Verify that the following requirements are met for the NetBackup server:

- The NetBackup server software is installed and operational on the NetBackup server.
  See the NetBackup Installation Guide.
  Every NetBackup server includes the NetBackup client software by default. Therefore, you can use NetBackup for SQL Server on a NetBackup server or client (if NetBackup for SQL Server is supported on that platform).
- Make sure that you configure any backup media that the storage unit uses. The number of media volumes that are required depends on several things:
  - The devices that are used and storage capacity of the media
  - The sizes of the databases that you want to back up
  - The amount of data that you want to archive
  - The size of your backups
  - The frequency of backups or archives
  - The length of retention of the backup images

  See the NetBackup Administrator's Guide, Volume I.

Verify that the following requirements are met for the NetBackup clients:

- The NetBackup client software is installed on the computer that has the databases you want to back up.
  If the database is clustered, you must use the same version of NetBackup on each node in the cluster.

- To use the new features that are included in NetBackup for SQL Server in NetBackup 8.1.1, you must upgrade your NetBackup for SQL Server clients to NetBackup 8.1.1. The NetBackup media server must use the same version as the NetBackup for SQL Server client or a higher version than the client.

Requirements for using NetBackup for SQL Server in a NetBackup cluster

If you plan to use NetBackup for SQL Server on a NetBackup server configured in a NetBackup cluster, verify the following requirements:

- NetBackup supports your cluster environment.
  See the Software Compatibility List (SCL).
- The NetBackup server software is installed and configured to work in a NetBackup cluster.
  See the NetBackup Installation Guide.
  See the NetBackup Clustered Master Server Administrator's Guide.
- The NetBackup client software is installed and operational on each node to which NetBackup can failover.
- A valid license for NetBackup for SQL Server must exist on each node where NetBackup server resides.

About the license for NetBackup for SQL Server

The NetBackup for SQL Server agent is installed with the NetBackup client software. No separate installation is required. A valid license for the agent must exist on the master server.

More information is available on how to add licenses.

See the NetBackup Administrator’s Guide, Volume I.

For a NetBackup cluster, a valid license for NetBackup for SQL Server must exist on each node where NetBackup server resides.

Chapter 3

Instance Management for SQL Server Intelligent Policies

This chapter includes the following topics:

About Instance management for a SQL Server Intelligent Policy

Instance management displays the instances that it discovers in the **Applications > Microsoft SQL Server** node of the NetBackup Administration Console. Any instances you add manually are also displayed in this node. Once the instances are registered, you can build a SQL Server Intelligent Policy.

Command line for DBAs

DBAs can run `nbsqladm` on a NetBackup client if the backup administrator authorizes a specific user and host on the master server. See the `nbsqladm` description in the NetBackup Commands Reference Guide.

Hosts that use multiple NICs

When NetBackup discovers a SQL Server host that uses multiple NICs, it adds the host with its NetBackup client name. If you installed the NetBackup client with the public interface name, further configuration is required before you can perform backups.

See "Configuring the NetBackup client with the private interface name" on page 173.

See "Configuring backups of SQL Server when you have multiple NICs (SQL Server Intelligent Policies)" on page 174.

SQL Server cluster in Instance Management

For a SQL Server cluster or a SQL Server cluster with multiple NICs, NetBackup adds a single entry or one instance to instance management. For a cluster, the host name for that instance is the virtual name of the SQL Server cluster. For a SQL Server cluster with multiple NICs the host name for that instance is the public virtual name of the SQL Server cluster. Further configuration is required both these environments before you can perform backups.

See "Configuring backups of clustered SQL Server instances (SQL Server Intelligent Policy) " on page 147.

See "Configuring backups of a SQL Server cluster when you have multiple NICs (SQL Server Intelligent Policies)" on page 177.

About credentials used with SQL Server Intelligent Policy

To protect an instance group or instance with a SQL Server Intelligent Policy, you must register the group or instance with credentials. Refer to Table 3-1 to determine the best option for your environment.

SQL Server instances must be registered with Windows credentials that have the proper permissions to perform backup and restore operations. Intelligent Policy supports Windows authentication and Windows Active Directory authentication. It does not support Mixed Mode or SQL Server authentication. Credentials are not supported at the database level.

Table 3-1 Options to register credentials

Option to register credentials	Environment or configuration	Notes
Use these specific credentials	■ The SQL Server DBA provides the NetBackup administrator with the SQL Server user credentials. ■ The SQL Server DBA does not want the NetBackup services running as a privileged SQL Server user on the client.	(Recommended) Veritas recommends that you use this option to register credentials. See the section called "Requirements when you register instances using specific credentials" on page 26.
Use credentials that are defined locally on the client	■ The user account that installed NetBackup is already running as a SQL Server privileged account. ■ The SQL Server DBA does not want to provide credentials to register instances. ■ The NetBackup administrator does not have access to the SQL Server credentials.	The NetBackup services run as a privileged SQL Server user on the client. See the section called "Requirements when you register instances using locally defined credentials" on page 26.
Add to group and register using group credentials	You want to be able to do one or more of the following: ■ Logically group your instances in some way. ■ Use a particular tuning parameter to improve the performance for each of the instances in the group. ■ (Optional) Automatically register new instances and add them to a group.	A group's credentials can be configured to use a specific set of credentials (each instance uses the same credentials). Or a group can be configured to use locally defined credentials (each instance uses the credentials that are defined for that instance). See "Registering instances with an instance group" on page 36.
Command line	■ The DBA does not have access to the NetBackup Administration Console. ■ The NetBackup administrator does not have the credentials for SQL Server. ■ The DBA wants to maintain the SQL Server credentials independently of the backup administrator.	See the section called "Configuring credentials from the command line" on page 26.

Requirements when you register instances using specific credentials

The following requirements apply when you use the option **Use these specific credentials** to register an instance or instances in an instance group:

- The user must have the SQL Server “sysadmin” role.
- The user must be a member of the Windows Administrators group.
- The logon account for the NetBackup Client Service and the NetBackup Legacy Network Service can be either the SQL System administrator or Local System. The services do not have to use the same logon account.
  See “Configuring the NetBackup services for SQL Server backups and restores” on page 27.
- The logon account for the NetBackup Client Service and the NetBackup Legacy Network Service must have the privileges to **Impersonate a client after authentication** and **Replace a process level token**.
  See “Configuring local security privileges for SQL Server” on page 29.

Requirements when you register instances using locally defined credentials

When you use the option **Use credentials that are defined locally on the client** to register an instance or instances in an instance group, NetBackup uses the credentials for the user that installed NetBackup. The following requirements apply with this option:

- The user must have the SQL Server “sysadmin” role.
- The user must be a member of the Windows Administrators group.
- The logon account for the NetBackup Client Service and the NetBackup Legacy Network Service can be either the SQL System administrator or Local System. The services must use the same logon account.
  See “Configuring the NetBackup services for SQL Server backups and restores” on page 27.

Configuring credentials from the command line

To register an instance from the command line, the following configuration is required:

- The NetBackup administrator must authorize the `nbsqladm` command for a specific DBA or user on a specific host.
  On the NetBackup master server, use `nbsqladm` to authorize the user:
  `nbsqladm [-S master_server] -add_dba host_name user_name`

If you have multiple NICs, authorize the DBA using the private interface name of the SQL Server host. For a SQL Server cluster, authorize the DBA for each node in the cluster. (Do not authorize a DBA using the virtual name of the SQL Server cluster.) For the `-host` *`name`* provide one of the node names in the SQL cluster. For a SQL Server cluster with multiple NICs, authorize the DBA using the private interface name for each of the nodes in the SQL Server cluster.

- Once a DBA is authorized to use the `nbsqladm` command, the DBA can register instances with the local credentials (`-local_credentials`) or other specific credentials (`-user name -domain name`).

For complete details on the `nbsqladm` command, see the NetBackup Commands Reference Guide.

Registering instances when SQL Server hosts are clustered or use multiple NICs

When NetBackup discovers a SQL Server cluster, it adds a single entry to instance management. This instance represents all nodes in the cluster. The host name is the virtual name of the SQL Server cluster. When you register this instance NetBackup validates the credentials on the active node. The credentials must be valid for all nodes in the cluster.

When NetBackup discovers a SQL Server host that uses multiple NICs, it adds an entry using the NetBackup client name to instance management. If you installed the NetBackup client using the public interface name, you must configure the NetBackup client name as the private interface name. Then register the instance with its private interface name. For a SQL Server cluster that uses multiple NICs, add and register the instance with the private virtual name of the SQL Server cluster.

See "Configuring the NetBackup client with the private interface name" on page 173.

Validation of credentials

More details are available troubleshooting the validation of credentials.

See "Troubleshooting credential validation with instance management" on page 243.

Configuring the NetBackup services for SQL Server backups and restores

NetBackup uses the NetBackup Client Service and the NetBackup Legacy Network Service to access the SQL Server when it performs backups and restores. With the proper configuration, these services can log on with the Local System account or another account that has the necessary privileges.

The logon account for the services requires the following:

- The SQL Server “sysadmin” role.
- If you want to use Local System for the logon account, the requirements depend on the SQL Server version:
  - For SQL Server 2008, the sysadmin role is automatically applied to the NT AUTHORITY\SYSTEM and BUILTIN\Administrators groups.
  - For SQL Server 2012 and later, you must first apply the sysadmin role manually to the NT AUTHORITY\SYSTEM or the BUILTIN\Administrators group.
- For a SQL Server cluster, configure the NetBackup services on each node in the cluster.
- Additional requirements depend on the credentials option you chose to register the instance(s).
  - For **Use these specific credentials**, the NetBackup services can use the Local System logon account. If you want to use a different logon account, it must have the privileges to **Impersonate a client after authentication** and **Replace a process level token**.
    See “Configuring local security privileges for SQL Server” on page 29.
    Both services can use the same logon account or separate logon accounts.
  - For **Use credentials that are defined locally on the client**, the logon account for the services can be either the SQL System administrator or Local System. Both services must use the same logon account.
- For VMware backups, different configuration is required for logon account for the services.
  See “Configuring the NetBackup services for VMware backups that protect SQL Server” on page 106.

To configure the NetBackup services for SQL Server backups and restores

1. Log on to the Windows host with the account that has the sysadmin role and any necessary local security privileges.
2. Open the Windows Services application.
3. Double-click the **NetBackup Client Service** entry.
4. Click the **Log On** tab.

5 Confirm that **Local System account** or a SQL Server administrator account is configured.

If you use the setting **Use credentials that are defined locally on the client** to register instances, both services must use the same logon account. If you use the setting **Use these specific credentials** to register instances, the services can use the same logon or separate logon accounts.

6 Click **OK**.

7 Double-click the **NetBackup Legacy Network Service** entry.

8 Click the **Log On** tab.

9 Confirm that **Local System account** or a SQL Server administrator account is configured.

If you use the setting **Use credentials that are defined locally on the client** to register instances, both services must use the same logon account. If you use the setting **Use these specific credentials** to register instances, the services can use the same logon or separate logon accounts.

10 Click **OK**.

11 If you selected a different logon account, restart the services.

See "About credentials used with SQL Server Intelligent Policy" on page 24.

Configuring local security privileges for SQL Server

When you use the option **Use these specific credentials** to register an instance, the account you use requires certain local security privileges. These privileges are necessary since the NetBackup for SQL Server agent logs on as the SQL Server user when it accesses data.

Note: This configuration applies to local security privileges only. For domain-level privileges, contact your domain administrator.

To configure the local security privileges

1 Open the **Local Security Policy**.

2 Click **Local Policies**.

3 In the **User Rights Assignment**, add the account to the following policies:

- **Impersonate a client after authentication**

- **Replace a process level token**

4 Run the group policy update command (group policy update) for this change to take effect:

```
gpupdate /Force
```

5 If the NetBackup Client Service and the NetBackup Legacy Network Service use this account to log on, restart these services.

See "About credentials used with SQL Server Intelligent Policy" on page 24.

About the NetBackup Discovery Service

By default, this service reports to the master server when it finds instances of applications. However, the user can turn off discovery for a specific client, with the `bpsetconfig` utility. See the `REPORT_CLIENT_DISCOVERIES` option in the NetBackup Administrator's Guide.

Viewing the SQL Server instances and instance groups in instance management

In the **NetBackup Administration Console**, expand the **Applications** > **Microsoft SQL Server** node. Under that node are the following subnodes:

- **All Instances**
  This node includes a complete list of all SQL Server instances that NetBackup discovers or that you manually added.
  To immediately discover any new instances that you added to your environment since the last discovery ran, choose **Actions > Discover Instances**.
  See the section called "Discovering instances on demand" on page 31.
- **Instance Groups**
  This node displays an instance groups that you created. You can use instance groups to organize instances and to register all instances in the group with a single set of credentials.

Table 3-2 describes the properties for instances and instance groups.

Table 3-2 Properties in Instance Management

Column	Description
Cluster Type	Cluster Type
Edition	The SQL Server edition.

Table 3-2 Properties in Instance Management *(continued)*

Column	Description
Host	The name of the host on which the instance resides. This host name is the name used for the backup in the NetBackup catalog. For a host that uses a multi-interface network connection (multi-NIC), NetBackup discovers and adds the host with the NetBackup Client name. If you installed the NetBackup client with the public interface name, you must also add and register the instance with its private interface name. Then add the instance with the private interface name to the backup policy. For a SQL Server cluster, the host name is the virtual name of the SQL Server cluster. If you have SQL Server cluster with a multi-NIC, you must also add and register the instance with the private name of the virtual SQL Server.
Instance Group	The instance group name that this instance is part of. This field is blank if the instance does not belong to an instance group.
Instance Name	The instance name. For a SQL Server cluster, NetBackup adds a single entry or one instance to instance management. The host name for that instance is the virtual name of the SQL Server cluster.
OS	The operating system of the host.
Policies	The name of any Intelligent Policies in which the instance or the instance groups appear. Legacy policies (using clients and batch files) are not reflected here.
Registered	Reflects the date and time when the instance was registered with valid credentials. This field is blank if the instance is not registered.
Release	The SQL Server release name.
SP	The SQL Server service pack number.
State	Active - The instance is available for backup by NetBackup. Inactive - This instance is inactive and cannot be backed up. This state implies that a NetBackup administrator purposely marked the instance as inactive in NetBackup. For example, if the instance is under maintenance.
Version	The SQL Server version number.

Discovering instances on demand

Since the discovery process does not run continually, any SQL Server instances that you add to your environment are not immediately discovered and added to the NetBackup database. The following procedure describes how to start NetBackup Discovery to find new instances.

To discover the SQL Server instances that you added after the last discovery

1. Open **Applications > Microsoft SQL Server > Instances**.
2. From the **Actions** menu select **Discover Instances**.

About instance actions

Table 3-3 describes the actions or operations that you can perform on SQL Server instances from the **Actions** menu.

Table 3-3 Instance actions

Action	Description
New > Instance	Manually adds an instance to instance management. See "Manually adding a SQL Server instance in instance management" on page 40.
Properties	Displays the instance properties. See "Registering a SQL Server instance" on page 34.
Register	Registers an instance. See "Registering a SQL Server instance" on page 34.
Delete	Deletes an instance from instance management. You cannot delete an instance that is part of a policy. First, delete the instance from the **Instances and Databases** tab in the policy.
Activate	Makes an instance that you deactivated in NetBackup available for backup.
Deactivate	Makes an instance inactive in NetBackup so it is excluded from a backup. For example, if the instance is under maintenance.
Remove from Group	Removes an instance from the instance group to which it was added. The instance is registered individually with the same credentials it had when it was a member of the instance group.
New Group with Instances	Adds one or more instances to a new group. If you previously registered an instance, the group credentials are applied to the instance and replace the previous credentials that you configured. See "Adding an instance to an instance group" on page 37.
Auto Registration	See "About instance group actions" on page 33.
Discover Instances	Immediately discovers any new instances that you added to your environment since the last discovery process.

Table 3-3 Instance actions *(continued)*

Action	Description
Instance Cleanup	This option lets you configure NetBackup to automatically clear orphaned instances from instance management. Orphaned instances are the instances that were discovered at one time but were never registered. To enable instance cleanup, select **Clean up after**. Then select how often (**days**) that you want NetBackup to perform instance cleanup.

About instance group actions

Table 3-4 describes the actions or operations that you can perform on SQL Server instance groups from the **Actions** menu.

Table 3-4 Instance group actions

Action	Description
New > Instance Group	Creates a new instance group. See "Registering instances with an instance group" on page 36.
Properties	Displays the instance group properties.
Delete	Deletes an instance group. You cannot delete an instance group that is part of a policy. First, delete the instance group from the **Instances and Databases** tab in the policy.
Auto Registration	Configures an instance group to automatically register newly discovered instances. NetBackup adds newly discovered instances to the instance group that you choose. All new instances are automatically registered and use the credentials setting for the group. Only one instance group can be configured for automatic registration. See "Registering instances automatically" on page 38. See "About credentials used with SQL Server Intelligent Policy" on page 24.
Discover Instances	Immediately discovers any new instances that you added to your environment since the last discovery process.
Instance Cleanup	See "About instance actions" on page 32.

About registering SQL Server instances

All instances that you want protected as part of a SQL Server Intelligent Policy must be registered with credentials. These credentials must have certain privileges.

See "About credentials used with SQL Server Intelligent Policy" on page 24.

Veritas recommends that you use the Applications utility in the NetBackup Administration Console to register instances (**NetBackup Management > Applications > Microsoft SQL Server**). If preferred, the NetBackup administrator can also authorize a DBA to register instances on a specific host.

Instances can be registered in one of the following ways:

- Manually, for individual instances.
  In the Applications utility, the user selects and individually registers newly discovered instances.
  See "Registering a SQL Server instance" on page 34.
- Manually, by adding instances to an instance group.
  In the Applications utility, the user creates an instance group and adds instances to the group. Instances use the credentials setting configured for the group.
  See "Registering instances with an instance group" on page 36.
- Automatically, by configuring an instance group to automatically register newly discovered instances.
  In the Applications utility, the user creates an instance group and configures the group for automatic registration. Newly discovered instances are automatically added to the group and registered. Instances use the credentials setting configured for the group.
- Manually, with the `nbsqladm` command.
  The NetBackup administrator can authorize a DBA to use the `nbsqladm` command to register instances on a specific host.
  See "Authorizing a DBA to register instances with the `nbsqladm` command" on page 39.

Registering a SQL Server instance

This topic describes how to register a SQL Server instance manually in the Applications utility. For additional registration options, see the following topics:

See "Registering instances with an instance group" on page 36.

See "About configuration of SQL Server backups with multiple NICs" on page 171.

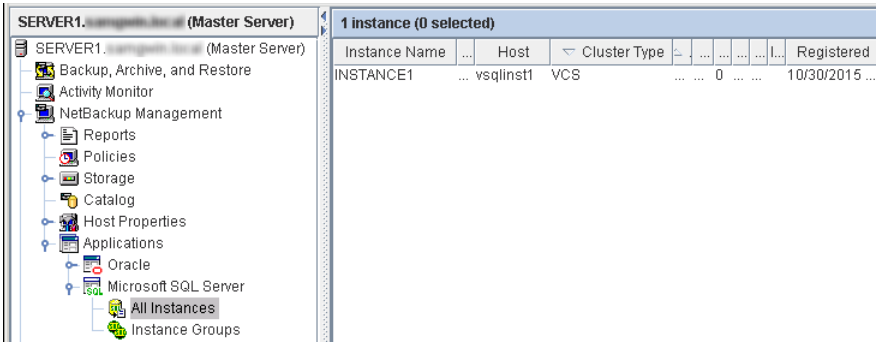
See "Authorizing a DBA to register instances with the `nbsqladm` command" on page 39.

To register a SQL Server instance

1. In the NetBackup Administration Console, in the left pane, expand **NetBackup Management > Applications > Microsoft SQL Server**.
2. Click **All Instances**.

   The right pane displays a list of instances. Instances that have previously been registered show a date and time in the **Registered** column.
3. Select the instance(s) that you want to register.

   For a SQL Server cluster, NetBackup adds a single entry or one instance to instance management. The host for that instance is the virtual name of the SQL Server cluster.



4. Choose **Actions > Register**.

   The **Edit Instance** dialog box appears.
5. To add credentials, click **Edit**.
6. In the **Register Instance** dialog box, select the credentials you want to use.

   The user account that is associated with these credentials must have the proper permissions to perform SQL Server backups and restores and register instances. More information is available to help determine which option best applies for your environment.

   See “About credentials used with SQL Server Intelligent Policy” on page 24.
7. Click **OK**.

   NetBackup validates the credentials, marks the instance as registered, and adds the instance to the NetBackup database. NetBackup requests detailed information about the instance from the NetBackup client and displays it in the **Microsoft SQL Server > Instances** node.

For a SQL Server cluster, NetBackup validates the credentials on the active node. The credentials must be valid for all nodes in the cluster.

If validation fails, a message displays. The user has the following options:

- Click **No** and enter different credentials. More information is available about validation failure.
  See "Troubleshooting credential validation with instance management" on page 243.
- Click **Yes** to save the credentials and add the instance, despite the validation failure. In this case, the instance is marked as registered even though the validation fails. NetBackup cannot successfully protect this instance without valid credentials.

8 In the right pane of the NetBackup Admin Console, review the **Registered** column to see that the instance is now registered.

9 Continue with any other instances that you want to register.

Registering instances with an instance group

Instance groups provide the following benefits when you create SQL Server policies:

- When you add an instance group to a policy, that single policy can back up many instances.
- You can configure an instance group to automatically add newly discovered instances to the group, registering instances on the fly. See "Registering instances automatically" on page 38.
- All the instances in the group use the same credentials setting. If you select the setting **Use these specific credentials**, you only need to enter those credentials once.
- In the Applications utility, you can easily see which policies protect which instance groups.

The following procedure describes how to create an instance group to which you can add instances.

To create an instance group

1 In the NetBackup Administration Console, in the left pane, expand **NetBackup Management > Applications > Microsoft SQL Server**.

2 Right-click **Instance Groups** and select **New Instance Group**.

3 Provide an **Instance Group Name**.

4 Select the credentials you want to use.

This user account must have certain privileges. More information is available to help determine which option best applies for your environment. See "About credentials used with SQL Server Intelligent Policy" on page 24.

5 Click **OK**.

6 To add instances to the group you created, see the following topic.

See "Adding an instance to an instance group" on page 37.

Adding an instance to an instance group

This topic describes how to add an instance to an instance group. Instances in a group all use the same credentials setting.

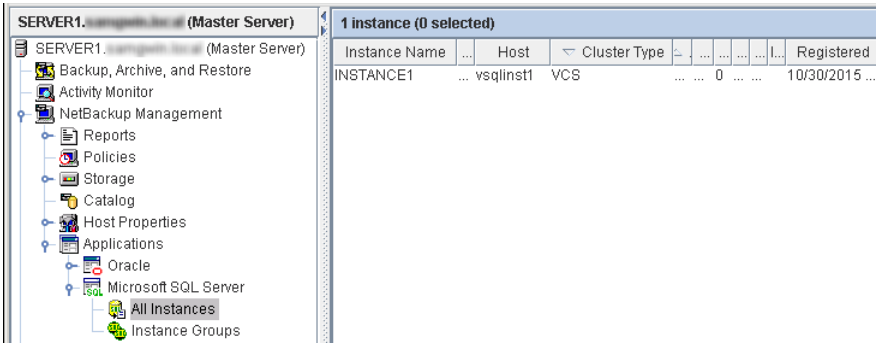
To add an instance to an instance group

1 In the NetBackup Administration Console, in the left pane, expand **NetBackup Management > Applications > Microsoft SQL Server**.

2 Click **All Instances**.

3 Select one or more instances that you want to add to an instance group.

For a SQL Server cluster, NetBackup adds a single entry or one instance to instance management. The host name for that instance is the virtual name of the SQL Server cluster.



4 From the **Actions** menu, select **Register**.

5 In the **Register Instance** dialog box, click **Add to group and register using group credentials**.

6 From the **Instance Group** list, select the instance group to which you want to add the instance(s).

7 Click **OK**.

If you previously registered an instance, its credentials are automatically changed to the group credentials setting. NetBackup validates the group credentials for the instance(s).

If the validation fails, you can choose to save the group or enter different credentials. See "Troubleshooting credential validation with instance management" on page 243. The backup of an instance fails if the credentials are not valid for that instance.

Registering instances automatically

With automatic registration, NetBackup adds newly discovered instances to the instance group that you choose. Only one instance group can be configured for automatic registration. All new instances are automatically registered and use the credentials setting for the group.

Note: Any instances that were discovered before this instance group was created are not automatically added to the group.

To register instances automatically

1 In the NetBackup Administration Console, in the left pane, expand **NetBackup Management > Applications**.

2 If necessary, create an instance group.

3 Click **Microsoft SQL Server** and choose **Actions > Auto Registration**.

4 In the **Automatic Registration** dialog box, select **Automatically register newly discovered instances**.

5 From the **Instance Group** list, select the instance group to which you want to add newly discovered instances.

6 Click **OK**.

To validate the credentials for the instances in the group, see the following topic.

See "Validating instance group credentials" on page 39.

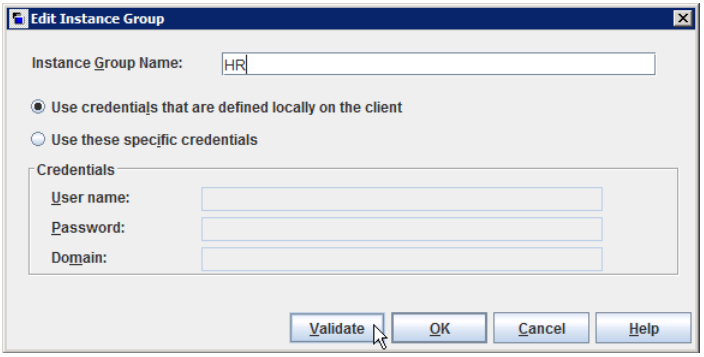
Validating instance group credentials

The following procedure describes how to validate the credentials for an instance group. Credentials are not validated when an instance is registered automatically. You should periodically validate the credentials for the instances in the group.

If you add an instance for a SQL Server cluster to an instance group, note that NetBackup validates the credentials on the active node. The credentials must be valid for all nodes in the cluster.

To validate group credentials

1. Select the instance group.
2. Choose **Actions > Properties**.
3. In the **Edit Instance Group** dialog box, and click **Validate**.



If the validation fails, you can choose to save the group or enter different credentials. The backup of an instance fails if the credentials are not valid for an instance.

See "Troubleshooting credential validation with instance management" on page 243.

Authorizing a DBA to register instances with the `nbsqladm` command

The NetBackup administrator can authorize a DBA to use the `nbsqladm` to register instances if the DBA wants to manage SQL Server credentials independently. From the master server the NetBackup administrator can control the list of users and hosts that can run `nbsqladm` on the NetBackup client.

For example, the NetBackup administrator can authorize the user john_smith on host winserver.domain.com with the following command:

```
nbsqladm -add_dba winserver.domain.com john_smith
```

From the NetBackup client, winserver.domain.com, john_smith can register and manage instances. For example, the DBA can register an instance with local credentials as follows:

```
nbsqladm -S NBUmaster1 -register_instance hr_city1
- host winserver.domain.com -local_credentials
```

More information on the `nbsqladm` command is available. See the NetBackup Commands Guide.

Manually adding a SQL Server instance in instance management

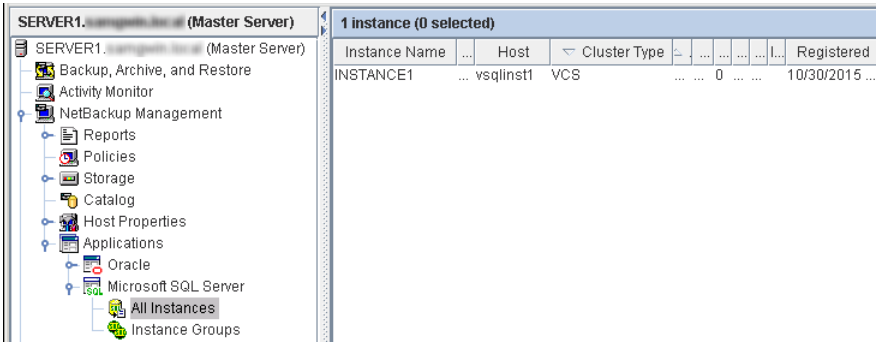
Newly discovered SQL Server instances on clients are automatically added to the NetBackup database. However, you may not want to wait for the discovery service to discover a new instance. In this case you can add an instance manually.

To manually add a SQL Server instance in instance management

1. In the NetBackup Administration Console, in the left pane, expand **NetBackup Management > Applications > Microsoft SQL Server**.
2. Right-click **All Instances** and select **New instance**.

3 Provide the **Instance Name** and **Host**.

For a SQL Server cluster or multi-NIC environment, add one entry to instance management. For a cluster, the host name is the virtual name of the SQL Server cluster. You do not need to add each node in the cluster to instance management. For a multi-NIC environment, the host name is the private interface name of the SQL Server host or of the virtual SQL Server.



4 Click **Edit** to provide credentials and register the instance.

See "Registering a SQL Server instance" on page 34.

You may omit credentials when you add a new instance to the NetBackup database. The instance is marked as unregistered and the **Registered** column in the right pane is empty. NetBackup cannot protect any instances that are not registered.

Note: If necessary, contact the SQL Server database administrator for the correct credentials. If the NetBackup administrator authorizes it, the DBA can also manually add the instance using the `nbsqladm`. This action is useful if the DBA does not share the credentials with the backup administrator.

Chapter 4

Configuring SQL Server backups with SQL Server Intelligent Policy

This chapter includes the following topics:

- Backing up read-write filegroups

About SQL Server Intelligent Policies

A SQL Server Intelligent Policy lets you create a single policy to protect multiple SQL Server instances or the databases in an instance. These instances can be spread over multiple clients. You can select SQL Server instances for a policy from a list of instances that are automatically discovered in the NetBackup environment.

The SQL Server Intelligent Policy includes the following criteria:

- Storage unit and media to use
- Policy attributes
- Backup schedules: Full, differential-incremental, transaction log
- Instances, databases in an instance, or instance groups to back up
- Backup selections: Whole database, filegroups, or files

NetBackup offers the following ways to configure a SQL Server policy:

- The Policy Configuration Wizard of the NetBackup Administration Console: The wizard guides you through the setup process by automatically choosing the best values for most configurations.
- The SQL Server Policy utility of the NetBackup Administration Console: This utility contains all the settings and parameters you need to create or change a SQL Server Intelligent Policy.

See “About configuring SQL Server backups with SQL Server Intelligent Policy” on page 43.

About configuring SQL Server backups with SQL Server Intelligent Policy

This topic describes the steps to configure your environment so you can perform SQL Server backups with a SQL Server Intelligent Policy.

Table 4-1 Steps to configure SQL Server backups that use a SQL Server Intelligent Policy

Step	Action	Description
Step 1	Verify that you have a supported SQL Server configuration.	See the Application/Database Agent Compatibility List.
Step 2	Ensure that requirements are met for the NetBackup server and the SQL Server software.	See "NetBackup server and client requirements" on page 21.
Step 3	Configure the NetBackup services for SQL Server backups and restores.	See "Configuring the NetBackup services for VMware backups that protect SQL Server" on page 106.
Step 4	Configure the local security privileges for SQL Server	See "Configuring local security privileges for SQL Server" on page 29.
Step 5	Register the SQL Server instances.	See "About Instance management for a SQL Server Intelligent Policy" on page 23. See "Registering a SQL Server instance" on page 34.
Step 7	Configure a SQL Server Intelligent policy.	See "Adding a new SQL Server Intelligent Policy" on page 44.
Step 8	If your SQL Server is clustered, you must configure the mappings for distributed application restores.	Configure these mappings in the **Distributed Application Restore Mapping** host property on the master server. See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.
Step 9	If your SQL Server is clustered, you must review the auto-discovered mappings for the hosts in your environment.	Approve each valid **Auto-Discovered Mapping** that NetBackup discovers in your environment. Perform this configuration in the **Host Management** properties on the master server. See "Reviewing the auto-discovered mappings in Host Management" on page 66.

Adding a new SQL Server Intelligent Policy

This topic describes how to add a new backup policy for a SQL Server database.

To add a new NetBackup for SQL Server Intelligent Policy

1. Log on to the master server as administrator (Windows) or root (UNIX).
2. Start the NetBackup Administration Console.

3. If your site has more than one master server, choose the one on which you want to add the policy.
4. In the left pane, expand **NetBackup Management** and select **Policies**.
5. Select **Actions > New > Policy**.
6. In the **Add a New Policy** dialog box, in the **Policy name** box, type a unique name for the new policy.
7. Click **OK**.
8. In the **Change Policy** dialog box, in the **Policy type** list, select **MS-SQL-Server**.
9. Complete the entries on the **Attributes** tab.

   See "About policy attributes" on page 45.
10. Add other policy information as follows:
    - Choose to protect instances or instance groups.
      If you choose the instances option, you can select either individual instances or databases.
      See "Adding instances to a policy" on page 50.
      See "Adding databases to a policy" on page 51.
      See "Adding instance groups to a backup policy" on page 57.
    - Add schedules.
      See "About schedule properties" on page 46.
    - (Optional) Select the specific filegroups or files that you want to back up. By default, NetBackup backs up an entire database.
      See "Adding filegroups or files to the backup selections list" on page 55.
    - (Optional) Make changes to any tuning parameters.
      See "About tuning parameters for SQL Server backups" on page 58.
11. When you have completed the policy configuration, click **OK**.

About policy attributes

With a few exceptions, NetBackup manages the policy attributes set for a database backup like a file system backup. Other policy attributes vary according to your specific backup strategy and system configuration.

Table 4-2 describes some of the policy attributes available for a NetBackup for SQL Server policy. For more information on policy attributes, see the NetBackup Administrator's Guide, Volume I.

Table 4-2 Policy attribute descriptions for NetBackup for SQL Server policies

Attribute	Description
Policy type	Determines the types of clients that can be backed up with the policy. For SQL Server databases, select the policy type MS-SQL-Server.
Limit jobs per policy	Sets the maximum number of instances that NetBackup can back up concurrently with this policy.
Compress	Enables the compression of backups by NetBackup. If you enable NetBackup compression, do not enable SQL Server compression. For more information on advantages and disadvantages of compression, see the NetBackup Administrator's Guide, Volume I.
Keyword phrase	Although you can create a keyword phrase for MS-SQL-Server policies, NetBackup for SQL Server does not record this information with the backup image.
Snapshot Client and Replication Director	This group contains the options that enable backups with Snapshot Client and Replication Director. See "About NetBackup Snapshot Client for SQL Server" on page 113. See "Configuring a VMware policy to protect SQL Server using Replication Director to manage snapshot replication" on page 109.

About schedule properties

This topic describes how to configure certain schedule properties for SQL Server Intelligent Policies. Other schedule properties vary according to your specific backup strategy and system configuration. Additional information about other schedule properties is available in the NetBackup Administrator's Guide, Volume I.

Table 4-3 describes how the schedule properties affect a SQL Server Intelligent Policy.

Table 4-3 Description of schedule properties

Property	Description
Type of backup	Specifies the type of backup that this schedule can control. The selection list shows only the backup types that apply to the policy you want to configure. See "Schedule backup types for SQL Server Intelligent Policies" on page 48.

Table 4-3 Description of schedule properties *(continued)*

Property	Description
Schedule type	You can schedule a backup in one of the following ways: ■ **Frequency** **Frequency** specifies the period of time that can elapse until the next backup operation begins on this schedule. For example, assume that the frequency is 7 days and a successful backup occurs on Wednesday. The next full backup does not occur until the following Wednesday. Typically, incremental backups have a shorter frequency than full backups. The frequency can be hours, days, or weeks. For transaction log backups, the frequency can also be minutes. ■ **Calendar** The **Calendar** option lets you schedule the backup operations that are based on specific dates, recurring week days, or recurring days of the month.
Retention	Specifies a retention period to keep backup copies before they are deleted. The retention period for a schedule controls how long NetBackup keeps records of when scheduled backups occurred. Set the time period to retain at least two full backups of your database. In this way, if one full backup is lost, you have another full backup to restore. The type of schedule you select affects the retention period as follows: ■ Frequency-based scheduling Set a retention period that is longer than the frequency setting for the schedule. For example, if the frequency setting is set to one week, set the retention period to be more than one week. The NetBackup scheduler compares the latest record of the backup schedule to the frequency of that backup schedule. This comparison determines whether a backup is due. So if you set the retention period to expire the record too early, the scheduled backup frequency is unpredictable. However, if you set the retention period to be longer than necessary, the NetBackup catalog accumulates unnecessary records. When NetBackup expires a backup image it does not notify SQL Server. Use SQL Server to periodically delete expired backup sets from the SQL Server repository. ■ Calendar-based scheduling The retention period setting is not significant for calendar-based scheduling.
Media multiplexing	Multiplexing is useful if you have many simultaneous backups using the same tape drive. However, it can interfere with SQL Server recovery due to how SQL Server requests streams during restore. In most cases, Veritas does not recommend multiplexing multiple SQL Server streams from the same backup to a single tape. See “Configuring multiplexed backups of SQL Server” on page 266.

Schedule backup types for SQL Server Intelligent Policies

The **Type of backup** attribute specifies the type of backup that the schedule controls.

Refer to the following guidelines when you configure schedules:

- If you require a 24-hour schedule for transaction log backups, create a separate policy for the transaction log backup schedule.
  See the section called "Configuring high-frequency transaction log backups" on page 49.
- The backup operation is skipped for a specific database if the database recovery model is not supported for the selected backup type. See the section called "Schedules and unsupported recovery models" on page 49.
- If a differential backup runs and a full backup does not already exist for the database or filegroup, NetBackup can convert the backup to a full backup. Similarly, NetBackup can convert transaction log backups if a full backup does not already exist for the database. Enable this behavior with the options **Convert differential backups to full (when no full exists)** or **Convert log backups to full (when no full exists)**.
  For snapshot backup policies, you must create a **Full Backup** schedule for NetBackup to successfully convert differential backups to full backups.
  See "About tuning parameters for SQL Server backups" on page 58.
- If you have read-only filegroups, follow the instructions for backing up read-only and read-write filegroups separately.
  See "Reducing backup size and time by using read-only filegroups" on page 164.

Table 4-4 shows the backup types you can specify.

Table 4-4 Schedule backup types for SQL Server Intelligent Policies

Backup type	Description
Full Backup	A complete backup of the database that contains all of the data files and the log file. (Note that a full backup does not truncate the transaction log.)
Differential Incremental Backup	A backup of the changed blocks since the last full backup. If you configure a differential incremental backup, you must also configure a full backup.

Table 4-4 Schedule backup types for SQL Server Intelligent Policies *(continued)*

Backup type	Description
Transaction Log backup	A backup of the active and the inactive portion of the transaction log. By default, the inactive portion is truncated after a successful backup. A transaction log backup can only be performed against a database that is configured to run in the full recovery model. You can choose to turn off truncation in the **Microsoft SQL Server** tab. See the section called "Configuring high-frequency transaction log backups" on page 49. If you want to configure transaction log backups to run at a high-frequency, review the recommendations. See "Configuring the number of jobs allowed for backup operations" on page 74.

Configuring high-frequency transaction log backups

Consider the following when you configure transaction log backups:

- Create a dedicated storage unit for transaction log backup images.
- Create a separate policy for transaction log backups and for full (and differential) backups.
- Configure the number of jobs that are allowed for backup operations. See "Configuring the number of jobs allowed for backup operations" on page 74.

Schedules and unsupported recovery models

NetBackup skips database backups in certain situations. The first case is if the database recovery model for a database does not support the selected backup type. For example, the simple recovery model does not allow transaction log backups. The second case is for the master database, which is skipped for any backups other than full database backups. To back up the master database, you must have a full backup schedule and select **Whole database** in the backup selections. Specifically, the master database is skipped for the following types of backups: differential, filegroup, filegroup differential, file, and transaction log.

In these cases, NetBackup skips the backup of the database, but continues with the backup of the other databases that are protected by the policy. The backup completes with a status 0 and the job details indicate that the database was skipped.

Example backup schedules for a policy

Table 4-5 shows an example of the schedules you can create for a single SQL Server Intelligent Policy.

Table 4-5 Examples of backup schedules

Schedule	Frequency	Backup window
Full Backup	Weekly	Sunday 12 hours
Differential Incremental Backup	Daily	Monday - Saturday 2 hours in the evening
Transaction Log backup	Per your RTO and RPO	Sunday - Saturday 24 hours **Note:** When the full or the differential schedule for the policy runs, the transaction log backup does not run until that schedule completes.

Adding instances to a policy

This topic describes how to add instances to a policy when you choose the **Protect instances** option. You can also add individual databases to the same policy.

See "Adding databases to a policy" on page 51.

To add instances a policy

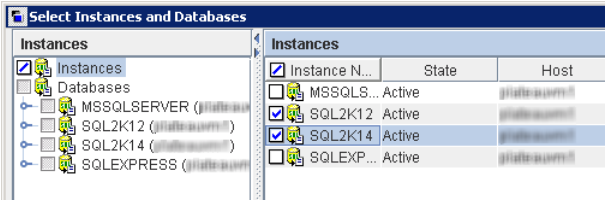
1. On the **Instances and Databases** tab, click **Protect instances**.
2. Click **New**.

   The **Select Instances and Databases** dialog box displays all the instances that you registered in the Applications utility.

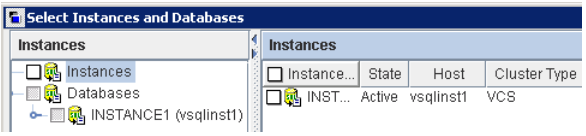
   A description of the properties in this list is available.

   See "Instance properties" on page 54.
3. In the left pane, select the **Instances** node.

4 In the right pane, check the check box next to each instance that you want to add to the list.

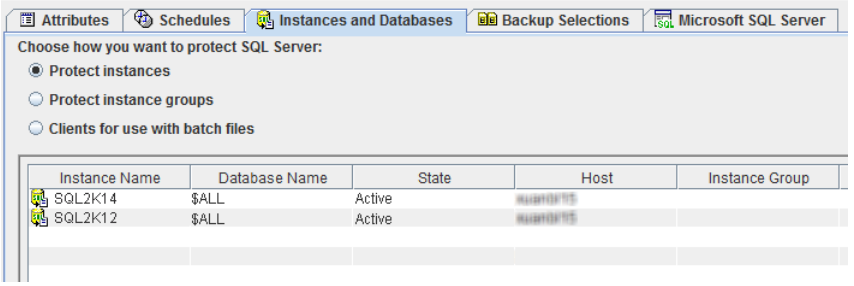


Note: Note that for a SQL Server cluster, there is only one entry that is displayed for the cluster. This entry represents all nodes in the cluster; the host is the virtual name of the SQL Server cluster.



5 Click **OK**.

The objects you select in the backup selections list apply only to the instances or databases that you add to the list on this tab.



Adding databases to a policy

This topic describes how to add databases to a policy when you choose the **Protect instances** option. You can also add instances to the same policy.

See “Adding instances to a policy” on page 50.

You cannot mix instances and instance groups. If you create a policy with instances or databases and later select the **Protect instance groups** option, the instances or databases are deleted from the policy.

To add databases to a policy

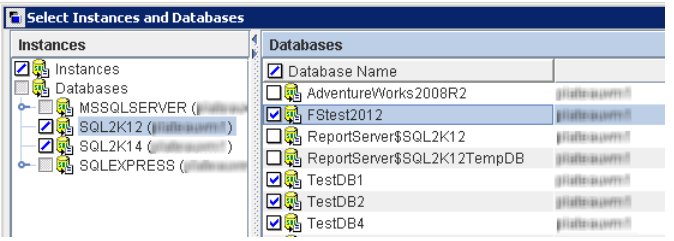
1. On the **Instances and Databases** tab, click **Protect instances**.
2. Click **New**.

   The **Select Instances and Databases** dialog box displays all the instances that you registered in the Applications utility.

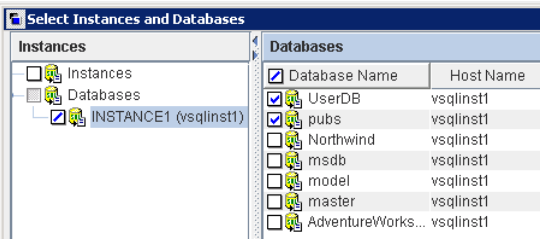
   See “Instance properties” on page 54. for a description of the properties for the instances and databases that are displayed in this list.
3. In the left pane, expand the **Databases** node and select the instance that contains the databases that you want to protect.

4 In the right pane, check the check box next to each database that you want to add to the list.

When you select individual databases, you must manually add any new databases in your environment to a policy. In this case, NetBackup does *not* dynamically create a list of databases at run-time.

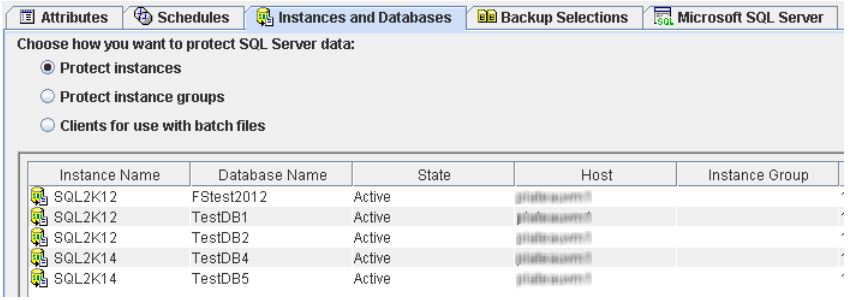


For databases that are hosted on a SQL Server cluster, the **Host Name** represents the virtual name of the SQL Server. (See the following figure.)



5 Click **OK**.

The objects you select in the backup selections list apply only to the instances or databases that you add to the list on this tab.



Instance properties

Table 4-6 describes the properties that you see for the instances and databases or instance groups that you add on the **Instances and Databases** tab.

Table 4-6 Instance properties on Instances and Databases tab

Field	Description
Instance Name	The name of the instance. For a SQL Server cluster, NetBackup adds a single entry or one instance to instance management. The host name for that instance is the virtual name of the SQL Server cluster.
Database Name	$ALL indicates that all databases for each instance are included in the backup. At backup time, NetBackup dynamically creates the list of databases to back up. This list reflects any new databases that you added to your SQL Server environment or any databases that you removed since you created the policy. When you select individual databases, this column displays the database names that you selected. If you add new databases to your environment, you must manually add these databases to a policy. In this case, NetBackup does *not* dynamically create a list of databases at run-time.
State	Active - The instance is available for backup by NetBackup. Inactive – This instance is inactive and cannot be backed up by NetBackup. This state implies that a NetBackup administrator purposely marked the instance as inactive. For example, if the instance is under maintenance
Cluster Type	For a clustered instance, indicates the type of cluster. For example, VCS or WSFC (Windows Server Failover Cluster).
Host	The name of the host on which the instance resides. This host name is the name used for the backup in the NetBackup catalog. For a host that uses a multi-interface network connection (multi-NIC), NetBackup discovers and adds the host with the NetBackup Client name. If you installed the NetBackup client with the public interface name, you must also add and register the instance with its private interface name. Then add the instance with the private interface name to the backup policy. For a SQL Server cluster, the host name is the virtual name of the SQL Server cluster.
Instance Group	Indicates the name of the instance group that the instance is a member of. This field is blank if the instance does not belong to an instance group.
Registered	Reflects the date and time when the instance was registered with valid credentials. This field is blank if the instance is not registered.

Backup Selections tab

On the **Backup Selections** tab choose the SQL Server object type to back up. Note that the displayed file names in the list are the logical names rather than the physical names of the files.

Whole database	By default, the **Whole database** option is selected. NetBackup protects the entire database when you select this option.
Filegroups	Click **Browse** to browse for individual filegroups. Click **New** to manually add the name of a filegroup.
Files	Click **Browse** to browse for individual files. Click **New** to manually add the name of a file.

Adding filegroups or files to the backup selections list

This topic describes how to browse for the filegroups or the files that you want to add to the backup selections list.

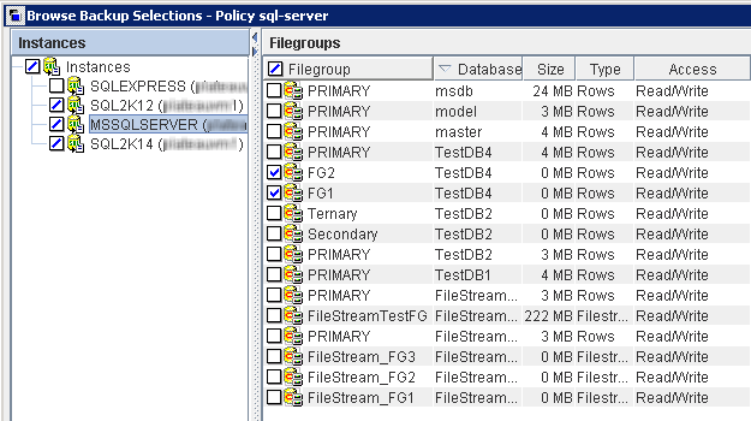
If you select specific databases and specific filegroups or files in a backup policy, NetBackup reports any unsuccessful filegroup or file backups differently than if you select an entire instance (`DATABASE $ALL`). Consider the following scenarios:

- Scenario 1 - For `SQLINSTANCE1` (`DATABASE $ALL` or all the databases), back up the filegroups `FG1`, `FG2`, and `FG3`. If NetBackup cannot back up `FG1`, `FG2`, or `FG3`, NetBackup skips the backup of the filegroup for that database. The parent job completes with a status 0.
- Scenario 2 - For `DATABASEA` and `DATABASEC` in `SQLINSTANCE1`, back up the filegroups `FG1`, `FG2`, and `FG3`. If NetBackup cannot back up any of these filegroups for `DATABASEA` or `DATABASEC`, the parent job completes with a status 2. The job details indicate that one or more of the filegroups that you selected were not backed up.

To add filegroups or files to the backup selections list

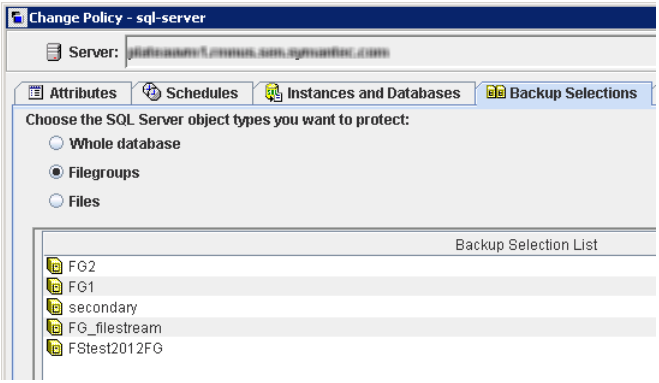
1. Open the policy you want to edit or create a new policy.
2. Select **Filegroups** or **Files**.
3. Click **Browse**.
4. In the left pane, select an instance to view the filegroups or files that it contains.

5 In the right pane select the filegroups or files.



6 Click **OK** to add the filegroups or files that you selected to the backup selections list.

Note: When you add a filegroup or file to the backup selections list, NetBackup backs up that object for all databases in the policy that contain a filegroup or file with that name.



Manually adding files or filegroups to the backup selections list

This topic describes how to manually add SQL Server database filegroups or files to the backup selections list.

To manually add files or filegroups to the backup selections list

1. Open the policy you want to edit or create a new policy.
2. Click the **Backup Selections** tab.
3. Select one of the SQL Server object types to back up:
   - **Filegroups**
   - **Files**
4. Click **New**.
5. Type the name of a filegroup or file and then click **Add**.

   Repeat this step to add any other filegroups or files.
6. Click **OK** to add the list of objects you selected to the backup selections list.
7. Click **OK** to save the backup selections list.

Adding instance groups to a backup policy

This topic describes how to add instance groups to a SQL Server Intelligent Policy. You cannot mix instances and instance groups. For example, if you create a policy with instance groups and later select the **Protect instances** option, the instance groups are deleted from the policy.

To add instance groups to a SQL Server Intelligent policy

1. Open the policy you want to edit or create a new policy.
2. On the **Instances and Databases** tab, click **Protect instance groups**.
3. Click **New**.

   The **Select Instance Group** dialog box displays all instance groups that you created in the Applications utility.
4. Select the instance group(s) you want to add and click **OK**.

   The list of instance groups that is displayed here controls the instances you can browse and select from when you create the backup selections list.

   To see a list of all the instances in the group, select the instance group and click **Preview Instances**.

About tuning parameters for SQL Server backups

The **Microsoft SQL Server** tab contains the tuning parameters that can improve the performance of your backups. These settings, and other factors that affect performance, are discussed in the following topic.

See "About NetBackup for SQL performance factors" on page 70.

Caution: Do not enable multiplexing if the policy is also configured with multiple stripes. Restores fail when both multiplexing and multiple stripes are configured for a backup policy.

Table 4-7 Tuning parameters for SQL Server backups

Field	Description
Number of backup stripes	This option divides the backup operation into multiple concurrent streams. A stream corresponds to a job in the activity monitor. For example, if the value is **3**, each database is backed up using three jobs. This configuration applies in any situation in which SQL Server dumps data faster than your tape drive is capable of writing. The default value for this option is 1. Range is 1-32. See "Configuring multistriped backups of SQL Server" on page 76.
Client buffers per stripe	(Stream-based backups only) This option affects buffer space availability. NetBackup uses this parameter to decide how many buffers to allocate for reading or writing each data stream during a backup operation. By allocating a greater number of buffers, you can affect how quickly NetBackup can send data to the NetBackup master server. The default value for this option is 2, which allows double buffering. You may get slightly better performance by increasing this value to a higher value. Range is 1-32.
Maximum transfer size	(Stream-based backups only) This option is the buffer size used by SQL Server for reading and writing backup images. Generally, you can get better SQL Server performance by using a larger value. This option can be set for each backup operation. Calculated as 64 KB * 2^`MAX_TRANSFER_SIZE`. It ranges in size from 64 KB to 4 MB. The default is 4 MB.
Backup block size	This option applies to stream-based backups only. Sets the incremental size that SQL Server uses for reading and writing backup images and can be set for each backup operation. Calculated as 512 bytes * 2^`BLOCK_SIZE`. The value for this option ranges from 0.5 KB to 64 KB. The default is 64 KB.

Table 4-7 Tuning parameters for SQL Server backups *(continued)*

Field	Description
Parallel backup operations	This option is the number of backup operations to start simultaneously, per database instance. Range is 1-32. The default is 1. You may need to configure other options when you configure two or more parallel backup operations. See "Configuring the number of jobs allowed for backup operations" on page 74.

Microsoft SQL Server checksum	Choose one of the following options for SQL Server backup checksums:
	
	■ None. Disables backup checksums.
	■ To verify the checksums before the backup, choose one of the following options. Note that these options impose a performance penalty on a backup or restore operation.
	  ■ Continue on error. If the backup encounters a verification error, the backup continues.
	  ■ Fail on error. If the backup encounters a verification error, the backup stops.

Use Microsoft SQL Server compression	Enable this option to use SQL Server to compress the backup image. If you enable SQL Server compression, do not enable NetBackup compression. SQL Server compression is not supported for snapshot backups.
Skip unavailable (offline, restoring, etc.) databases	NetBackup skips any database with a status that prevents NetBackup from successfully backing up the database. These statuses include offline, restoring, recovering, and emergency mode, etc. NetBackup skips the backup of the unavailable database, but continues with the backup of the other databases that the policy includes. The backup completes with a status 0 and the job details indicate that the database was skipped. See "Schedule backup types for SQL Server Intelligent Policies" on page 48.
Copy-only backup	This option allows SQL Server to create an out-of-band backup so that it does not interfere with the normal backup sequence. The default value is unchecked except for full database Instant Recovery backups. See "Using copy-only snapshot backups to affect how differentials are based" on page 121.
Skip read-only file groups	This option excludes any filegroups that are read-only from the backup. The resulting backup is a partial image because the image does not contain all filegroups. The partial image contains data from the read-write filegroups and data from the primary filegroup. This option applies only to the **Whole database** backup selection. See "Backing up read-only filegroups" on page 61. See "Backing up read-write filegroups" on page 62.

Table 4-7 Tuning parameters for SQL Server backups *(continued)*

Field	Description
Convert differential backups to full (when no full exists)	If no previous full backup exists for the database or filegroup, then NetBackup converts a differential backup to a full backup. The agent checks to determine if a full backup exists for each database. If no previous full backup exists, a differential backup is converted to a full as follows: ■ If you select a database for a differential backup, the backup is converted to a full database backup. If the **Skip read-only file groups** option is selected the backup is converted to a full read/write filegroup backup. ■ If you select a filegroup for a differential backup, NetBackup does the following: ■ If the filegroup is the default database filegroup, NetBackup converts the backup to a full filegroup backup. ■ If the filegroup is a secondary filegroup and a backup of the primary filegroup does not exist, NetBackup converts the backup to a partial full database backup. This backup contains the selected filegroup and default filegroup. ■ If the filegroup is a secondary filegroup and a backup of the primary filegroup does exist, NetBackup converts the backup to a full filegroup backup of the selected filegroup. ■ For snapshot backup policies, you must create a **Full backup** schedule for NetBackup to successfully convert differential backups to full backups. Note: NetBackup only converts a differential backup if a full backup was never performed on the database or filegroup. If a full backup does not exist in the NetBackup catalog but SQL Server detects an existing full LSN, NetBackup performs a differential backup and not a full. In this situation, you can restore the full backup with native tools and any differentials with the NetBackup MS SQL Client. Or, if you expired the backup in NetBackup, you can import the full backups into the NetBackup catalog. Then you can restore both the full and the differential backups with the NetBackup MS SQL Client.
Truncate logs after backup	This option backs up the transaction log and removes the inactive part of the transaction log. This option is enabled by default.

Table 4-7 Tuning parameters for SQL Server backups *(continued)*

Field	Description
Convert log backups to full (when no full exists)	If no previous full backup exists for the database, then NetBackup converts a transaction backup to a full backup. This option also detects if a full recovery database was switched to the simple recovery model and back to the full recovery model. In this scenario, the log chain is broken and SQL Server requires a differential backup before a subsequent log backup can be created. If NetBackup detects this situation, the backup is converted to a differential database backup. Note: NetBackup only converts a transaction log backup if a full backup was never performed on the database. If a full backup does not exist in the NetBackup catalog but SQL Server detects an existing full LSN, NetBackup performs a transaction log backup and not a full. In this situation, you can restore the full backup with native tools and any differentials and log backups with the NetBackup MS SQL Client. Or, if the backup was expired by NetBackup, you can import the full backups into the NetBackup catalog. Then you can restore the full, differential, and log backups with the NetBackup MS SQL Client.

Backing up read-only filegroups

When you separate read-only and read-write filegroups in your backup strategy, you can reduce total media usage and the total time you spend on backup operations. To back up read-only filegroups you must first create a separate policy for this type of backup. You can also verify that all read-only filegroups are backed up.

See "Backing up read-write filegroups" on page 62.

See "Reducing backup size and time by using read-only filegroups" on page 164.

See "Viewing SQL Server read-only backup sets" on page 217.

To back up read-only filegroups

1. Create a new policy to protect read-only filegroups.
2. Select the policy attributes.

   See "About policy attributes" on page 45.
3. Create a **Full** backup schedule and set the **Retention** level to **Infinite**.

   All read-only filegroups must be included in some combination of full, or individual filegroup and file backups. You only need to perform this backup one time. See "About schedule properties" on page 46.

4 Choose to protect instances or instance groups.

See “Adding instances to a policy” on page 50.

See “Adding instance groups to a backup policy” on page 57.

5 On the **Backup Selections** tab, select **Filegroups**.

See “Adding filegroups or files to the backup selections list” on page 55.

6 Select the filegroups you want to back up.

7 When you complete the policy configuration, click **OK**.

8 Back up the read-only filegroups.

9 If necessary, confirm all read-only groups are backed up by viewing the read-only backup set.

Backing up read-write filegroups

When you separate read-only and read-write filegroups in your backup strategy, you can reduce total media usage and the total time you spend on backup operations. More information is available on backing up read-only filegroups.

See “Backing up read-only filegroups” on page 61.

Note: Immediately back up any filegroup when you change it from read-write to read-only.

To back up read-write filegroups

1 Create a new policy or open the policy you want to configure.

2 Select the policy attributes.

See “About policy attributes” on page 45.

3 Create a **Full Backup**, **Differential Incremental Backup**, and **Transaction Log backup** schedule.

See “About schedule properties” on page 46.

4 On the **Instances and Databases** tab, choose to **Protect instances**.

5 Add the instances or the databases that contain the read-write filegroups.

See “Adding instances to a policy” on page 50.

6 On the **Backup Selections** tab, select **Whole database**.

7 Click the **Microsoft SQL Server** tab.

8 Check **Skip read-only file groups**.

See “About tuning parameters for SQL Server backups” on page 58.

9 When you have completed the policy configuration, click **OK**.

Chapter 5

Configuring NetBackup for SQL Server

This chapter includes the following topics:

Configuring mappings for restores of a distributed application, cluster, or virtual machine

This configuration is required for restores of a SQL Server cluster or a SQL Server availability group (AG). Configure these mappings in the Distributed Application Restore Mapping host property on the master server.

To configure mappings for restores of a distributed application, cluster, or virtual machine

1. On the master server, open the NetBackup Administration Console.
2. Select **NetBackup Management > Host Properties > Master Servers**.
3. In the right pane, double-click on the master server.

4 Select **Distributed Application Restore Mapping**.

5 Click **Add**.

6 Provide the name of the application host and the name of the component host.

See Example entries for SQL Server

Example entries for SQL Server

Table 5-1 Example entries for SQL Server

Environment	Application host	Component host
FCI (cluster with two nodes)	Virtual name of the SQL Server cluster	Physical name of *Node 1*
	Virtual name of the SQL Server cluster	Physical name of *Node 2*
AG (primary and secondary)	WSFC name	Primary name
	WSFC name	Secondary name
AG with an FCI (primary FCI and secondary FCI)	WSFC name	Primary FCI name
	WSFC name	Secondary FCI name
	Virtual name of the SQL Server cluster	Physical name of *Node 1*
	Virtual name of the SQL Server cluster	Physical name of *Node 2*
VMware	VM display name, VM BIOS UUID, or VM DNS name (Primary VM identifier other than **VM hostname**)	Host name of the VM

Reviewing the auto-discovered mappings in Host Management

In certain scenarios, a NetBackup host shares a particular name with other hosts or has a name that is associated with a cluster. To successfully perform backups and restores with NetBackup for SQL Server, you must approve each valid **Auto-Discovered Mapping** that NetBackup discovers in your environment. These mappings appear in the Host Management properties on the master server. You can also use the `nbhostmgmt` command to manage the mappings. See the Security and Encryption Guide for more details on Host Management properties.

Examples of the configurations that have multiple host names include:

- A host is associated with its fully qualified domain name (FQDN) and its short name or its IP address.
- If the SQL Server is clustered, the host is associated with its node name and the virtual name of the cluster.

Auto-discovered mappings for a cluster

In a SQL Server cluster environment, you must map the node names to the virtual name of the cluster if the following apply:

- If the backup policy includes the cluster name (or virtual name)
- If the NetBackup client is installed on more than one node in the cluster
  If the NetBackup Client is only installed on one node, then no mapping is necessary.

To approve the auto-discovered mappings for a cluster

1. In the NetBackup Administration Console, expand **Security Management > Host Management**.
2. At the bottom of the **Hosts** pane, click the **Mappings for Approval** tab.

   The list displays the hosts in your environment and the mappings or additional host names that NetBackup discovered for those hosts. A host has one entry for each mapping or name that is associated with it.

   For example, for a cluster with hosts `client01.lab04.com` and `client02.lab04.com`, you may see the following entries:

Host	Auto-discovered Mapping
client01.lab04.com	client01
client01.lab04.com	clustername
client01.lab04.com	clustername.lab04.com
client02.lab04.com	client02
client02.lab04.com	clustername
client02.lab04.com	clustername.lab04.com

3. If a mapping is valid, right-click on a host entry and click **Approve**.

   For example, if the following mappings are valid for `client01.lab04.com`, then you approve them.

Auto-discovered Mapping	Valid name for
client01	The short name of the client
clustername	The virtual name of the cluster
clustername.lab04.com	The FQDN of the virtual name of the cluster

4 When you finish approving the valid mappings for the hosts, click on the **Hosts** tab at the bottom of the **Hosts** pane.

For hosts `client01.lab04.com` and `client02.lab04.com`, you see **Mapped Host Names/IP Addresses** that are similar to the following:

Host	Mapped Host Names/IP Addresses
client01.lab04.com	client01.lab04.com, client01, clustername, clustername.lab04.com
client02.lab04.com	client02.lab04.com, client02, clustername, clustername.lab04.com

5 If you need to add a mapping that NetBackup did not automatically discover, you can add it manually.

Click on the **Hosts** tab, then right-click in the **Hosts** pane and click **Add Shared or Cluster Mappings**. For example, provide the name of the virtual name of the cluster. Then click **Select Hosts** to choose the node names in the cluster to which you want to map that virtual name.

In Table 5-2, FCI is a SQL Server failover cluster instance. AG is an availability group. WSFC is Windows Server Failover Cluster.

Table 5-2 Example mapped host names for SQL Server environments

Environment	Host	Mapped Host Names
FCI (cluster with two nodes)	Physical name of *Node 1*	Virtual name of the SQL Server cluster
	Physical name of *Node 2*	Virtual name of the SQL Server cluster
AG (primary and secondary)	Primary name	WSFC name
	Secondary name	WSFC name
AG with an FCI (primary FCI and secondary FCI)	Primary FCI name	WSFC name
	Secondary FCI name	WSFC name
	Physical name of *Node 1*	Virtual name of the SQL Server cluster
	Physical name of *Node 2*	Virtual name of the SQL Server cluster

Auto-discovered mappings for a SQL Server cluster in a multiple NIC environment

If you have a SQL Server cluster in a multi-NIC environment, you need to approve each valid **Auto-Discovered Mapping** for the hosts in that environment. You must map the virtual name of the SQL Server cluster on the private network to the private name of each SQL Server cluster node.

To approve the auto-discovered mappings for a SQL Server cluster in a multiple NIC environment

1. In the NetBackup Administration Console, expand **Security Management > Host Management**.
2. At the bottom of the **Hosts** pane, click the **Mappings for Approval** tab.

   The list displays the hosts in your environment and the mappings or additional host names that NetBackup discovered for those hosts. A host has one entry for each mapping or name that is associated with it.

   For example, for a cluster in a multi-NIC environment with hosts `client01-bk.lab04.com` and `client02-bk.lab04.com`, you may see the following entries:

Host	Auto-discovered Mapping
client01-bk.lab04.com	clustername-bk.lab04.com
client02-bk.lab04.com	clustername-bk.lab04.com

3. If a mapping is valid, right-click on a host entry and click **Approve**.

   For example, if following mapping is valid for `client01-bk.lab04.com`, then you approve it.

Auto-discovered Mapping	Valid name for
clustername-bk.lab04.com	The virtual name of the SQL Server cluster on the private network

4 When you finish approving the valid mappings for the hosts, click on the **Hosts** tab at the bottom of the **Hosts** pane.

For hosts `client01-bk.lab04.com` and `client02-bk.lab04.com`, you may see the following **Mapped Host Names/IP Addresses**.

Host	Mapped Host Names/IP Addresses
client01-bk.lab04.com	clustername-bk.lab04.com
client02-bk.lab04.com	clustername-bk.lab04.com

5 If you need to add a mapping that NetBackup did not automatically discover, you can add it manually.

Click on the **Hosts** tab, then right-click in the **Hosts** pane and click **Add Shared or Cluster Mappings**. For example, provide the name of the virtual name of the cluster. Then click **Select Hosts** to choose the hosts to which you want to map that virtual name.

Example mapped host names for a SQL Server cluster in a multi-NIC environment

Table 5-3 Example mapped host names for a SQL Server cluster in a multi-NIC environment

Host	Mapped Host Names
Private name of *Node 1*	Virtual name of the SQL Server cluster on the private network
Private name of *Node 2*	Virtual name of the SQL Server cluster on the private network

About NetBackup for SQL performance factors

Many factors can influence the backup performance, including your hardware environment and the settings in SQL Server and NetBackup. To optimize your system for SQL Server backups the first step is to tune your environment for standard backup operations. Details are provided in the NetBackup Backup Planning and Performance Tuning Guide. When this tuning is complete, you can adjust several things specific to SQL Server.

Note: Some of the factors are only applicable to SQL Server stream-based operations and have no affect on snapshot backups or restores.

For a SQL Server Intelligent policy, set these parameters in the policy, on the **Microsoft SQL Server** tab. For a backup batch file (legacy SQL Server policy) or for a restore batch file, configure these parameters in the NetBackup MS SQL Client interface. The parameters in the NetBackup client properties are saved for the session.

The following factors can affect performance:

- NetBackup for SQL buffer space parameters
- Stripes and parallel backup operations
- Shared memory usage
- Alternate buffer method
- Microsoft SQL Server checksum
- Instant data file initialization
- Using read-write and read-only filegroups

NetBackup for SQL buffer space parameters

The **Maximum transfer size**, **Backup block size**, and **Client buffers per stripe** can increase buffer space in SQL Server. SQL Server must have the available resources to support the increase of these values. Buffer space parameters are applicable for stream-based backups only.

The **Maximum transfer size** parameter can be set for each backup or restore operation. **Maximum transfer size** is the buffer size used by SQL Server for reading and writing backup images. Generally, you can get better SQL Server performance by using a larger value.

The **Backup block size** parameter can be set for each backup operation. For restore operations, NetBackup automatically chooses the same size that that was used for the backup. **Backup block size** is the incremental size that SQL Server uses for reading and writing backup images.

The **Client buffers per stripe** determines how many buffers to allocate for reading or writing each data stream during a backup or restore operation. Setting this factor to a value greater than **1** enables multi-buffer during data transfer. By allocating a greater number of buffers, you can affect how quickly NetBackup can send data to the NetBackup media server. Multi-buffer prevents short-term producer-consumer imbalances during a backup or restore operation. Although you can set the number of buffers as high as **32**, normally a value of **2** or **3** is sufficient.

Stripes and parallel backup operations

You can improve performance and throughput by increasing the backup stripes or parallel backup operations, depending on the size and number of databases.

Enabling multiple stripes (**Number of backup stripes**) is useful for larger databases when the performance gains outweigh the additional overhead necessary for the SQL Server agent to configure them. When protecting smaller databases, striping can decrease performance speed. In general, if the SQL Server instance only has a few large databases, the use of stripes improves performance. If the instance has numerous smaller databases, increasing the amount of **Parallel backup operations** is a better choice to improve performance. You can increase both stripes and parallel backup operations at the same time, but be careful not to overwhelm the system resources.

See "Configuring the number of jobs allowed for backup operations" on page 74.

Caution: Do not enable multiplexing if the policy is also configured with multiple stripes. Restores fail when both multiplexing and multiple stripes are configured for a backup policy.

Shared memory usage

Optimal performance is seen if you install NetBackup server on the same host as NetBackup for SQL Server. Also use shared memory for data transfer instead of sockets. Shared memory is the default for this configuration and is used unless you create a `install_path\NetBackup\NOSHM` file.

Alternate buffer method

NetBackup for SQL Server supports an alternate buffer method. It optimizes CPU usage by allowing NetBackup and SQL Server to share the same memory buffers without transferring data between them.

The alternate buffer method for backup and restore typically does not improve data transfer rate, only CPU utilization. A situation may occur in which the transfer rate is significantly degraded when alternate buffer method is in use. To improve the transfer rate set the **Maximum transfer size** for the backup to the maximum allowed, which is 4 MB.

About alternate buffer method with backup operations

This method is chosen automatically for backups if all of the following conditions apply:

- NetBackup shared memory is in use.
- The backup is stream-based.
- The backup is not multiplexed.
- The backup policy does not specify either NetBackup compression or NetBackup encryption.

- The NetBackup buffer size equals the SQL Server block size.
  The default NetBackup buffer size is 64 KB, but this value can be overridden in the following settings:
  install_path\NetBackup\db\config\SIZE_DATA_BUFFERS (for tape backups), or,
  install_path\NetBackup\db\config\SIZE_DATA_BUFFERS_DISK (for disk backups)
- NetBackup for SQL Server agent is started with the same account as the NetBackup Client Service.
  The backups that are initiated from an automatic backup policy are started with the NetBackup Client Service so the same account is already in use. However, you can start a SQL Server backup through NetBackup for SQL Server or through dbbackex. In this case, your logon account must be the same as the NetBackup Client Service account. Then your backups can be candidates for the alternate buffer method.

About alternate buffer method with restore operations

Conditions for backups require that you use the alternate buffer method. Restores also require that backups have been made with the alternate buffer method. You can verify that the alternate buffer method was used. Look for the words `Using alternate buffer method`, which appear in the `dbclient` log and the progress report.

Microsoft SQL Server checksum

You can choose to perform a checksum before you perform a backup. When this option is enabled, it imposes a performance penalty on a backup or restore operation.

For legacy backup policies, set the **Page verification** value when you create the script. For restore scripts, choose **Verify backup image, but don't restore** option when you create the script.

Instant data file initialization

When you restore a database, filegroup, or database file, SQL Server zeroes the file space before it begins the restore operation. This action can slow the total recovery time by as much as a factor of 2. To eliminate file initialization, run the MSSQLSERVER service under a Windows account that has been assigned the SE_MANAGE_VOLUME_NAME. For more information, see the SQL Server and the Windows documentation.

Using read-write and read-only filegroups

You can significantly reduce backup time and the storage media that is needed if you periodically back up only read-write filegroups. Then keep a single backup of read-only filegroups, which is retained infinitely. You can set the retention level in the schedule.

Configuring the number of jobs allowed for backup operations

When NetBackup starts a backup of SQL Server, a number of jobs are created. Depending on the policy configuration, additional jobs are created if you configure settings such as **Number of backup stripes** and **Parallel backup operations**. (For legacy policies, the equivalent settings are the **Stripes** setting and the `BATCHSIZE` keyword.)

You can increase or limit the number of jobs that are created. You can also control the number of jobs that are sent to the storage unit. Consider the following settings.

Limit jobs per policy	Sets the maximum number of instances that NetBackup can back up concurrently in each policy. This setting is configured in the policy attributes. See the NetBackup Administrator's Guide, Volume I.
Maximum jobs per client	In a policy, the maximum number of jobs per client that you want to allow. This setting applies to all clients in all policies. It is configured in the master server host properties on the **Global Attributes** node. See "Configuring the Maximum jobs per client setting" on page 75.
Maximum concurrent jobs	The maximum number of jobs that NetBackup can send to a storage unit at one time. This setting is configured in the storage unit properties. See the NetBackup Administrator's Guide, Volume I.
Maximum concurrent write drives	The number of tape drives that NetBackup can use at one time for jobs to this storage unit. This setting is configured in the storage unit properties. See the NetBackup Administrator's Guide, Volume I.

Configuring the Maximum jobs per client setting

The **Maximum jobs per client** specifies the maximum number of concurrent backups that are allowed per instance or database (Intelligent Policies). Each instance or database that is specified in the policy creates a new backup job. For legacy policies, this setting indicates the maximum that is allowed per client.

To configure the maximum jobs per client

1. In the left pane of the NetBackup Administration Console, expand **NetBackup Management > Host Properties**.
2. Select **Master Server**.
3. In the right pane, double-click the server icon.
4. Click **Global Attributes**.
5. Change the **Maximum jobs per client** value to the wanted value.

   The default is 1.

For Intelligent Policies, use the following formula to calculate a smaller value for the **Maximum jobs per client** setting:

Maximum jobs per client = *number_of_database_objects* X *number_of_streams* X *number_of_policies*

For legacy policies, use the following formula to calculate a smaller value for the **Maximum jobs per client** setting:

Maximum jobs per client = *number_of_streams* X *number_of_policies*

Refer to the following definitions:

number of database_objects	(Intelligent Policies) The number of databases, filegroups, or files that you want to back up in parallel.
number_of_streams	The number of backup streams between the database server and NetBackup. If striping is not used, each separate stream starts a new backup job on the client. If striping is used, each new job uses one stream per stripe.
number_of_policies	The number of policies of any type that can back up this client at the same time. This number can be greater than one. For example, a client can be in two policies to back up two different databases. These backup windows can overlap.

Configuring multistriped backups of SQL Server

SQL Server supports backups of databases through multiple data streams, which are called stripes. NetBackup stores each stripe as a separate image. The purpose of this feature is to speed up the rate of data transmission with the use of multiple tape devices.

Backup images can be written to more tapes than available drives. When you restore this type of backup image, in the restore batch file indicate the number of drives that are available.

See "Restoring multistreamed SQL Server backups" on page 98.

Caution: Do not enable multiplexing for a schedule that is also configured to backup with multiple stripes. Restores fail when multiplexing is enabled for a schedule that uses more than one stripe.

Configure the following to create a multistriped backup:

- In the backup policy, select the number of **Stripes** you want to use.
  For SQL Server Intelligent policy, configure this setting on the **Microsoft SQL Server** tab. For legacy SQL Server policies, configure the **Stripes** setting when you create the backup batch file.
- In the schedules for your policy, set **Media multiplexing** to **1** to disable multiplexing.
  For legacy SQL Server policies, disable multiplexing in the "Application Backup" schedule. When you disable multiplexing, during a restore all streams are made available simultaneously so the restore operations are successful.
- Ensure that the storage unit has as many drives as you want to have stripes.
- Configure backup schedules so that enough drives are available at the time you want to perform striped backups.

Performing a manual backup

After you configure the servers and clients in your environment, you can test the configuration settings with a manual backup. Perform a manual backup (or backups) with the automatic backup schedules you created. A description of status codes and other troubleshooting information is available.

See the NetBackup Status Codes Reference Guide.

See the NetBackup Logging Reference Guide.

To perform a manual backup

1. Log onto the master server as administrator (Windows) or root (UNIX).
2. Start the NetBackup Administration Console.
3. In the left pane, click **Policies**.
4. In the **All Policies** pane, select the policy you want to test.
5. Select **Actions > Manual Backup**.
6. Select the schedule that you want to use for the manual backup.
7. For SQL Server Intelligent Policies, select the databases or instances that you want to include for the manual backup. For legacy SQL Server policies, select the clients that you want to include for the manual backup.
8. To check the status of the backup, click **Activity Monitor** in the NetBackup Administration Console.

   The Activity Monitor and the script output indicate the status of the backup operation.

Chapter 6

Performing restores of SQL Server

This chapter includes the following topics:

- About restores of a database that contain full-text catalog
- Restoring multistreamed SQL Server backups

Starting the NetBackup MS SQL Client for the first time

This topic describes how to start the NetBackup MS SQL Client for the first time. For subsequent sessions, the agent remembers the information you provided.

To start the NetBackup MS SQL Client for the first time

1. If you use SQL Server integrated security, log on to the Windows host with the Windows account that has permissions to perform SQL Server backups and restores.
2. Open the NetBackup MS SQL Client.
3. When you are prompted to provide the logon parameters, click **OK**.
4. In the **SQL Server connection properties** dialog box, select the SQL Server host and instance that you want to log into.
5. If the SQL Server host and instance use standard or mixed security, provide the SQL Server user ID and password.
6. Click **Apply**.
7. Click **Close**.

Selecting the SQL Server host and instance

Use this procedure to set which SQL Server host and the instance that you want the NetBackup MS SQL Client to access.

(Legacy SQL Server policies) The user ID and password are only required if the host uses standard or mixed security. If applicable, you only need to provide these credentials when you first open the NetBackup MS SQL Client.

To select the SQL Server host and instance

1. Open the NetBackup MS SQL Client.
2. Select **File > Set SQL Server connection properties**.
3. In the **SQL Server connection properties** dialog box, from the **Host** drop-down list, select the SQL Server host.

   You can type a host name if it does not appear in the drop-down list. If you select a remote host and click **Apply**, the **Host type** is shown as "remote".

4 From the **Instance** drop-down list, select the SQL Server instance.

You can type an instance name if it does not appear in the drop-down list. You can designate the default instance either by setting the Instance box to <default> or to empty (no spaces).

5 Click **Apply** to save your changes.

6 Click **Close**.

Browsing for SQL Server backup images

This procedure describes how to browse for available backup images. When you have displayed the backup images you want, then follow the instructions for restoring a specific SQL Server object.

If you use a specific network interface for backups, see the following instructions.

See "Performing restores of SQL Server when you have multiple NICs" on page 176.

To browse for backup images

1 Change the host and instance you want to access.

See "Selecting the SQL Server host and instance" on page 79.

2 Select **File > Restore SQL Server objects**.

3 In the **Backup History Options** dialog box, select the **SQL Host** whose backup images you want to browse, or type its name.

See "Redirecting a SQL Server database to a different host" on page 95.

4 Indicate the **Source Client**, if applicable.

In most cases when you browse for backup images, you only need to specify the **SQL Host** name. When the NetBackup client name and the host name are different you also need to also provide the **Source Client** name. For example, if the NetBackup client name is the network interface name. For Intelligent Policies, you also need to indicate the **Source Client** if you add or register the instance with a host name that is different than the NetBackup client name.

5 Select the date range to search.

6 Click **OK**.

7 Continue with the applicable instructions for how to restore the object(s).

See "Restoring a SQL Server database backup" on page 84.

See "Staging a full SQL Server database recovery" on page 85.

See "Restoring SQL Server filegroup backups" on page 86.

See "Recovering a SQL Server database from read-write filegroup backups" on page 87.

See "Restoring SQL Server read-only filegroups" on page 87.

See "Restoring SQL Server database files" on page 88.

See "Restoring a SQL Server transaction log image without staging a full recovery" on page 88.

See "Performing a SQL Server database move" on page 89.

See "About performing a SQL Server page-level restore" on page 91.

Options for NetBackup for SQL Server restores

Table 6-1 describes the options that are available when you perform restores. These options appear in the **Restore Microsoft SQL Server Objects** dialog box after you select **File** > **Restore SQL Server objects**.

Table 6-1 Options for restore operations

Option	Description
Scripting	These scripting options are available for restoring from a database image: ■ **Restore selected object** Produce a script that performs a database restore. This script is the default option. ■ **Create a move template** Create a script template for moving the selected database. ■ **Restore read-only filegroups** Restore the most recent backup of every read-only filegroup. ■ **Create a page restore template** Create a template for restoring a database, filegroup, or file from the pages that are contained in the selected backup image. The Microsoft SQL Server service must have full access permission to the folder *install_path*\netbackup\dbext\mssql\temp. ■ **Verify backup image, but don't restore** This option is only available if the image was backed up with the page verification option. NetBackup processes the image for errors, but does not perform a restore.

Table 6-1 Options for restore operations *(continued)*

Option	Description
Use replace option	Restore with the SQL Server replace option.
Recovery	Specify one of the SQL Server recovery options. ■ **Not recovered** Use this option during a restore if additional backup images must be applied to the database following the current restore. When you use this option, the database is left in a loading state. ■ **Recovered** Select this option when restoring the last image in a restore sequence. After the recovery operation, the database is ready for use. If you do not select this option, the database is in an intermediate state, and is not usable. If **Recovered** is selected when an intermediate backup is applied, you cannot continue to restore backups; you must restart the restore operation from the beginning. ■ **Standby** Create and maintain a standby during a transaction log and database restore. This option requires a standby undo log, which by default is placed in *install_path*\NetBackup\logs\SQLStandBy\. The account that runs the Microsoft SQL Server service must have full access permission to the SQLStandBy folder. The database is placed in "standby" state following the restore.
Consistency check	Select the consistency check to be performed after the restore. Output from the consistency check is written to the SQL Server client progress log. You cannot select consistency checking unless the database is restored to the recovered state. If you select consistency checking for a staged recovery, then the check occurs following the last restore. ■ **None** Do not perform consistency checking. ■ **Full check, excluding indexes** Exclude indexes from the consistency check. If indexes are not checked, the consistency check runs significantly faster but is not as thorough. Only the data pages and clustered index pages for each user table are included in the consistency check. The consistency of the non-clustered index pages is not checked. ■ **Physical check only** Select this item to perform a low overhead check of the physical consistency of the SQL Server database. This option only checks the integrity of the physical structure of the page and record headers. It also checks the consistency between the pages' object ID and index ID and the allocation structures. ■ **Full check, including indexes** Include indexes in the consistency check. Any errors are logged. ■ **Check catalog** Check for consistency in and between system tables in the specified database.

Table 6-1 Options for restore operations *(continued)*

Option	Description
Page verification	**Note:** A performance penalty can happen when you use page verification. These options are available if the source object was backed up with torn page detection or checksum verification. ■ **Do not perform verification** Do not include page verification in the restore script. ■ **Perform verification** Include page verification in the restore script and stop the restore if an error is encountered.
Stage full recovery	Select this option to recover the database by using the recovery set that NetBackup found. If the transaction log that you select does not belong to a recovery set, this option is disabled.
Restore selected transaction log	Select this option to restore only the selected transaction log. If the transaction log that you select does not belong to a recovery set, this option is disabled.
Transaction log recovery options	This list contains the controls for you to restore a transaction log. You can restore the log to a point in time that precedes the time when the transaction log was dumped. The individual entries in this group are only enabled if you selected a transaction log backup. ■ **To point in time** Select this option to have the transaction log recovered to a point in time. ■ **To transaction log mark** Select this option to have the transaction log recovered to a transaction log mark. With this option, you must enter a transaction log mark name. ■ **To transaction log mark but after** Select this option to have the transaction log recovered to a transaction log mark but after a point in time. With this option, you must enter a transaction log mark name. ■ **Before transaction log mark** Select this option to have the transaction log recovered to a point before the occurrence of a transaction log mark. With this option, you must enter a transaction log mark name. ■ **Before transaction log mark but after** Select this option recover the transaction log to a point before the occurrence of a transaction log mark but after a point in time. With this option, you must enter a transaction log mark name.
Transaction log mark	This list is enabled if you selected a database transaction log for restore. The transaction log contains one or more transaction log marks, and you selected one of the following transaction log recovery options: ■ To transaction log mark ■ To transaction log mark but after ■ Before transaction log mark ■ Before transaction log mark but after

Table 6-1 Options for restore operations *(continued)*

Option	Description
YYYY, MM, DD, HH, MM, SS am, pm	Specify the time to which you want the transaction logs restored. These fields are only enabled if you selected one of the following transaction log recovery options: ■ To point in time ■ To transaction log mark but after ■ Before transaction log mark but after
Launch immediately	Start the restore operation immediately. **Launch immediately** is disabled if you are logged into a SQL Server instance that is not on the local host. If you generate a script for a non-local host, it must be executed on that host.
Save	Generate a script that can be started at a later time.
Restore	Start the restore or generate a restore script. This button is disabled if you have not selected any objects to restore.

Restoring a SQL Server database backup

This topic describes how to restore a database from a full database or differential database backup.

To restore a database backup

1. Browse for the backup images you want to restore.

   See "Browsing for SQL Server backup images" on page 80.

2. In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance.
3. Expand the database.
4. Select the database image that you want to restore, as follows:
   - To restore a full backup, select the image of the database backup.
   - To restore a full backup and a differential database backup, click the "+" and select a differential backup.
     The full backup is automatically selected when you select a differential.

5 Select the restore options.

To place the database in recovery mode so that it is immediately usable following the restore, select **Recovered** from the **Recovery** list. However, be aware that after the database is placed in recovered mode, you cannot update it with additional differential or transaction log backups.

See "Options for NetBackup for SQL Server restores" on page 81.

6 Click **Restore**.

7 To view the progress of the restore, select File > **View status**.

Staging a full SQL Server database recovery

This topic describes how to stage a full database recovery.

To stage a full database recovery

1 Browse for a backup image that contains the point in time to which you want to recover.

See "Browsing for SQL Server backup images" on page 80.

2 In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance.

3 Click the "+" next to the database that contains the transaction log backup you want to restore.

4 Select the transaction log image that includes the point in time from which you want to recover.

5 Select **Stage full recovery**.

Stage full recovery is enabled if a set of images exists that includes the transaction log image and that are adequate for staging a full database recovery. When you are viewing the properties of the transaction log, a Recovery Set tab appears.

The recovery set can include any combination of backup images that are sufficient for staging the full recovery. These can include full database, filegroup, and differentials.

6 Click **Restore**.

7 To view the progress of the restore, select File > **View status**.

Restoring SQL Server filegroup backups

This topic describes how to restore a backup of a filegroup. If your scheduled backups only include read-write filegroups, see the following topics.

See "Recovering a SQL Server database from read-write filegroup backups" on page 87.

See "Restoring SQL Server read-only filegroups" on page 87.

Note: If you attempt to restore a single differential backup without first restoring the preceding database backup file, SQL Server halts the load process. An error such as 4305 or 4306 is displayed. If you plan to restore a single differential, then you are responsible for first restoring the database backup file. You can avoid this problem by backing up the entire sequence of transaction logs. Also back up the differential backup and the backup file to the same NetBackup server. Then you can restore the entire sequence of backup objects.

Note: See "Staging a full SQL Server database recovery" on page 85.

To restore a filegroup backup

1. Browse for the backup images you want to restore.

   See "Browsing for SQL Server backup images" on page 80.

2. In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance and database.

3. Expand the filegroup and select a filegroup image to restore, as follows:

   - To restore a full backup, select the image of the filegroup backup.
   - To restore a differential filegroup backup, click the "+" next to the full backup and select the differential backup.

4. Select the restore options.

   See "Options for NetBackup for SQL Server restores" on page 81.

5. Click **Restore**.

   To view the progress of the restore, select File > **View status**.

Recovering a SQL Server database from read-write filegroup backups

NetBackup for SQL Server automatically generates the most efficient recovery path when you select a transaction log image for restore. The recovery path can be based on read-write filegroups if you use them in your backup strategy. After restoring the read-write filegroups, you can bring the database online without having to restore the read-only filegroups provided they are not damaged.

To recover a database from read-write filegroups

1. Browse for the backup images you want to restore.

   See "Browsing for SQL Server backup images" on page 80.

2. In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance.
3. Expand the database that contains the read-write filegroups you want to restore.
4. Select the transaction log backup.
5. Right-click the transaction log backup and select **Properties**.
6. On the **Recovery set** tab, verify that a complete backup set is available.
7. Click **OK**.
8. To begin the database restore, click **Restore**.

   After the restore completes the database is back online. However, you cannot recover the read-only filegroups until they are restored.

   See "Restoring SQL Server read-only filegroups" on page 87.

Restoring SQL Server read-only filegroups

This topic describes how to restore read-only filegroups.

To restore read-only filegroups

1. Browse for the backup images you want to restore.

   See "Browsing for SQL Server backup images" on page 80.

   Be sure that the start date for the Time Filter is early enough to include the timestamp of the earliest backup of the read-only filegroups.

2. In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance.

3 Select the database that contains the read-only filegroups you want to restore.

In the **Scripting** list, **Restore read-only filegroups** is selected.

The restore option is enabled if a full set of read-only filegroups is available.

4 Click **Restore**.

5 To view the progress of the restore, select File > **View status**.

Restoring SQL Server database files

This topic describes how to restore database files.

To restore a database file

1 Browse for the backup images you want to restore.

See "Browsing for SQL Server backup images" on page 80.

2 In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance and the database.

3 Expand the filegroup that contains the file you want to restore.

4 Expand the file.

5 Select the database file image that you want to restore.

6 Select the restore options.

See "Options for NetBackup for SQL Server restores" on page 81.

7 Click **Restore**.

To view the progress of the restore, select File > **View status**.

Restoring a SQL Server transaction log image without staging a full recovery

This topic describes how to restore a transaction log image without staging a full recovery.

To restore a transaction log without staging a full recovery

1 Browse for the backup images you want to restore.

See "Browsing for SQL Server backup images" on page 80.

2 In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance.

3 Select the transaction log image that you want to restore.

If a set of images exists that include the transaction log image and that are sufficient for staging a full database recovery, **Stage full recovery** is enabled. The properties of the transaction log include a **Recovery Set** tab.

4 Select **Restore only the transaction log that you selected**.

5 Click **Restore**.

To view the progress of the restore, select File > **View status**.

Performing a SQL Server database move

Note: NetBackup only supports a database move of a backup with FileStream enabled if the backup is stream-based.

A database move lets you use a full set of backup images to copy an existing database to a location under a different name. Database move operations can only be carried out when your selection includes a database image. This move can occur either when you directly select the database backup image, or when NetBackup finds a recovery set that contains a database backup image.

For information on redirected restores, see the following topic.

See "Redirecting a SQL Server database to a different host" on page 95.

To perform a database move

1 Browse for the backup images you want to restore.

See "Browsing for SQL Server backup images" on page 80.

2 In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance.

3 Select the database backup image that you want to restore.

4 From the **Scripting** list, select **Create a move template**.

When you create a move script, the capability to perform an immediate launch is disabled. You must edit the script to specify certain destination parameters.

5 Select the restore options.

See "Options for NetBackup for SQL Server restores" on page 81.

6 Click **Restore**.

7 Indicate a file name and click **Save**.

8 In the **Save Script As** dialog box, click **Yes** to open the template in Notepad.

9 Change the database name in the template to the name of the database to restore to.

For example, replace:

```
#  Replace the database name in the following line with the name of the database that you
#  want to move to. Also remove the hash mark <#> which precedes the keyword <DATABASE>.
#
# DATABASE "DatabaseA"
```

with:

```
#  Replace the database name in the following line with the name of the database that you
#  want to move to. Also remove the hash mark <#> which precedes the keyword <DATABASE>.
#
DATABASE "DatabaseB"
```

10 Change the path for the database files that you want to restore.

You must uncomment at least one file. For example, replace:

```
#  Replace the file path <C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DBA_FG1_File1.ndf>
#  with a new file path. Also remove the hash mark <#> which precedes the keyword <TO>.
#  The target of the MOVE keyword must be "DBA_FG1_File1".
MOVE  "DBA_FG1_File1"
#TO  "C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DBA_FG1_File1.ndf"
```

with:

```
#  Replace the file path <C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DBA_FG1_File1.ndf>
#  with a new file path. Also remove the hash mark <#> which precedes the keyword <TO>.
#  The target of the MOVE keyword must be "DBA_FG1_File1".
MOVE  "DBA_FG1_File1"
TO  "C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DBB_FG1_File1.ndf"
```

11 Change the database file path.

For example, replace:

```
#  Replace the file path <C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DatabaseA.mdf>
#  with a new file path. Also remove the hash mark <#> which precedes the keyword <TO>.
#  The target of the MOVE keyword must be "DatabaseA".
MOVE  "DatabaseA"
#TO  "C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DatabaseA.mdf"
```

with:

```
#  Replace the file path <C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DatabaseA.mdf>
#  with a new file path. Also remove the hash mark <#> which precedes the keyword <TO>.
#  The target of the MOVE keyword must be "DatabaseA".
MOVE  "DatabaseA"
TO  "C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DatabaseB.mdf"
```

12 Make similar changes to the template for any differential backups or transaction log backups you want to move.

13 When you finish modifying the template, save it.

14 To run the restore, select **File > Manage script files**, select the script you created, and click **Start**.

15 Click **Yes** to launch the restore.

To view the progress of the restore, select **File > View status**.

About performing a SQL Server page-level restore

Note: Page-level restores are only applicable for SQL Server legacy backup policies.

If a portion of a SQL Server database is corrupted due to hardware failure, you may be able to use page-level restore. Use page-level restore to recover only the pages that were corrupted. Page-level restore can reduce the total downtime if you only need to restore a relatively small number of pages. If many pages are corrupt, then a full database recovery may be faster.

When you select the page restore option, NetBackup for SQL Server creates a page restore template.

This template includes the following parts:

- A page restore operation that you can modify by inserting the IDs of the pages that you want to restore.

- A series of transaction log images for recovering the database to the current point in time.
- A tail-log backup and recovery operation, which is required to bring the database online.

About SQL page-level restore requirements and limitations

The following requirements and limitations exist when you perform SQL Server page-level restores:

- Pages can be restored from the following backup types: Database, filegroup, file, read-write filegroups, and partial database.
- Your SQL Server must use either the full or bulk-logged recovery model.
- SQL Server sometimes cannot recover the specific pages that you request if they contain critical information about the definition of the database itself. For example, you cannot use page-level restore for the first page in a database file. When you detect that page-level restore does not work, you need to use full database recovery.
- A maximum of 1000 pages can be recovered from a backup image through a page-level restore.

Performing SQL Server page-level restores

This topic describes how to perform page-level restores. Note that the Microsoft SQL Server service must have full access permission to the folder *install_path*\netbackup\dbext\mssql\temp.

To perform a page-level restore

1. Obtain a list of corrupt pages in the database.

   SQL Server Books Online suggests several methods for obtaining a list of corrupt pages. One of these methods is to run the command `DBCC checkdb` from the SQL Server Management Studio.

2. Browse for the backup images you want to restore.

   See "Browsing for SQL Server backup images" on page 80.

3. In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance.
4. Expand the database.
5. Select the database backup image that contains pages you want to restore.
6. From the **Scripting** list, select **Create a page restore template**.
7. Click **Restore**.

8. In the **Save Script As** dialog box, type a file name for the page restore script and click **Save**.
9. Click **Yes** to open the template in Notepad.
10. Edit the page first operation the page IDs that you want to replace.

    For example, replace:

```
#
# Create one or more page restore requests. These use the following format
#PAGE file-id:page-id
```

with

```
#
# Create one or more page restore requests. These use the following format
PAGE 1:14
PAGE 1:20
```

11. When you finish modifying the template, save it.
12. To run the restore, select **File > Manage script files**, select the script you created, and click **Start**.
13. Click **Yes** to start the restore.

Configuring permissions for redirected restores

Certain restore procedures or environments require that you configure permissions for redirected restores. This configuration allows a client to restore a backup that another client performed. See the NetBackup Administrator's Guide, Volume I for complete details on redirected restores.

You must configure the master server for redirected restores if you want to do the following:

- Redirect the restore of *ClientA* to *ClientB*
- Redirect a restore in a database mirroring environment to either of the database mirroring partners

You do not need to configure redirected restores for the following configurations. Instead these environments require that you configure the mappings for distributed application restores. You also need to review the auto-discovered mappings for the hosts in your environment.

- Restore databases in a SQL Server cluster to any of the nodes in the cluster
- Restore databases in an availability group (AG) to any of the nodes in the AG

- Restore clustered databases in a multi-NIC environment across the private interface

See "Reviewing the auto-discovered mappings in Host Management" on page 66.

See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.

To allow a specific client or host to perform a redirected restore

1. On the master server, create an `altnames` file for each client or host that you want to have permissions to perform redirected restores.

   For example, to give `HostB` permissions to redirect a restore, create the following file:

   On Windows:

   install_path`\NetBackup\db\altnames\HostB`

   On UNIX:

   `/usr/openv/netbackup/db/altnames/HostB`

2. In the `altnames` file, add the names of the hosts whose files the requesting client wants to restore.

   For example, assume that you want `HostB` to have permissions to redirect restores from `HostA`. Then add `HostA` to the `HostB` file.

To give a SQL Server host the permissions to restore backups in a multi-NIC environment

1. Create an `altnames` file with the private name of the host, for example `SQLHOST-NB`.

   On Windows:

   install_path`\NetBackup\db\altnames\SQLHOST1-NB`

   On UNIX:

   `/usr/openv/netbackup/db/altnames/SQLHOST1-NB`

2. In the `altnames` file, add the names of the hosts whose files the requesting client wants to restore.

   For example, assume that you want `SQLHOST1-NB` to have permissions to redirect restores from `SQLHOST2-NB`. Then add `SQLHOST2-NB` to the `SQLHOST1-NB` file.

Redirecting a SQL Server database to a different host

This topic describes how to redirect a backup to a client that is different from the client that performed the backup. You redirect a restore by performing a database move operation. NetBackup creates a template that you edit to indicate the host and location where you want to redirect the restore. The new location can be a different instance on the same host, a different host, or a different file path. The move operation also lets you restore the database under a different name than the original one. After you edit the template, select **File > Manage script files** to launch it.

Note: The destination host and instance of a move or restore operation is the one that you log into. For move or restore operations designate the source (or browse) host and the instance when you select **File > Restore SQL Server objects**.

To redirect a database to another location on a different host

1. Establish permissions for redirected restores on the master server.

   See "Configuring permissions for redirected restores" on page 93.

2. The server that backed up the database you want to restore must appear in the server list of the destination host. If the server is not in the list, add it.

   See "About selecting a master server " on page 96.

3. Select **File > Set SQL Server connection properties**.
4. From the **Host** list, select the host you want to restore to.
5. From the **Instance** list, select the database instance.

   To select the default instance, either select **<default>** or leave the field empty.

6. Click **Apply** and then **Close**.
7. Select **File > Set NetBackup client properties**.
8. In the **NetBackup client properties** dialog box, from the **Current NetBackup Server** list, select the NetBackup master server.

   This server contains the SQL Server backup images that you want to restore on the destination host. The clients must both use the same master server.

   See "About selecting a master server " on page 96.

9. Click **OK**.
10. Select **File > Restore SQL Server objects**.

11. In the **Backup History Options** dialog box, in the **SQL Host** list, select the host that has the database you want to restore.
12. Indicate the **Source Client**, if applicable.

    In most cases when you browse for backup images, you only need to specify the **SQL Host** name. When the NetBackup client name and the host name are different you also need to also provide the **Source Client** name. For example, if the NetBackup client name is the network interface name. For Intelligent Policies, you also need to indicate the **Source Client** if you add or register the instance with a host name that is different than the NetBackup client name.
13. Click **OK**.
14. Browse for the database that you want to move.
15. From the **Scripting** list, select **Create a move template**.
16. Click **Restore**.

    NetBackup prompts you to save the template.
17. In the **Save As** dialog box, enter a file name and click **Save**.
18. Click **Yes** to open the template.
19. Edit the template to designate the name that you want to use for the destination database. Also include the file paths that you want to use for each of the database files.

About selecting a master server

When you perform a move, the backup images must be available on the host machine that acts as the NetBackup master server for the destination host. If this server is contained in the server list of the destination host, you can select the current master server by selecting **File** > **Set NetBackup client properties**.

If the server is not in the server list of the destination host you must duplicate the images onto removable media (with a unique ID). Then transport that media to the master server that the destination host uses, and import the images to that server. After the images are imported, continue with the instructions for performing a move. A server may not appear in the server list because the server is remote or has access limitations.

See “Performing a SQL Server database move” on page 89.

Performing a restore of a remote SQL Server installation

You can use NetBackup for SQL Server to restore databases on a remote host. Generated batch files must be saved on the remote host. You can launch the operation from the local installation of NetBackup for SQL Server.

To perform a restore of a remote SQL Server installations

1. Select the host and instance you want to access.

   See "Selecting the SQL Server host and instance" on page 79.

2. Select **File > Restore SQL Server objects**.

3. Select the options for the operation.

   See "Options for NetBackup for SQL Server restores" on page 81.

   Save is enabled in the restore dialog box. **Launch immediately** is disabled because the generated script must be executed on the remote host that you are logged on to.

4. Click **Restore**.

5. In the **Save Script As** dialog box, navigate to the *install_path*\NetBackup\DbExt\MsSql\ folder on the remote host, and save the batch file there.

6. Run the operation from the local installation of NetBackup for SQL Server.

About restores of a database that contain full-text catalog

As of SQL Server 2008 and later, full-text search catalogs are included in the database backup and are restored as a part of the recovery process. When you restore a filegroup, NetBackup restores the full-text index files and other files in the filegroup. For additional information on backing up and restoring full-text catalogs and indexes, see the following Microsoft article.

https://msdn.microsoft.com/en-us/library/ms142511.aspx

If you want to back up and restore the full-text catalogs that you upgraded from SQL Server 2005, refer to the following Microsoft article:

https://msdn.microsoft.com/en-us/library/ms142490.aspx

Restoring multistreamed SQL Server backups

When you use the NetBackup MS SQL Client, backups using multiple stripes are automatically restored using the same number of stripes. Select the object you want to restore and NetBackup finds all of the related backups and restore them. Upon restore, all of the streams must also be available at the same time.

About conventional backups using multiple streams

If you specified multiple stripes for a non-snapshot backup, then the number of backup streams that you specified was created. NetBackup names these streams, for example:

```
juneberry.MSSQL7.COLE.db.pubs.~.7.001of003.20140908200234..C
juneberry.MSSQL7.COLE.db.pubs.~.7.002of003.20140908200234..C
juneberry.MSSQL7.COLE.db.pubs.~.7.003of003.20140908200234..C
```

To create your own batch file to restore a striped object, specify only the first stripe name with the NBIMAGE keyword. NetBackup for SQL Server finds the remaining ones automatically. More information is available about the backup names that are used for SQL Server objects

See "About using bplist to retrieve SQL Server backups" on page 227.

About snapshot backup methods using multiple streams

If you specified multiple stripes for any Snapshot Client backup, which streams the frozen image to tape, then NetBackup divides the number of component files equally among the number of stripes. If the number of files is less than the specified number of stripes, then the agent performs the backup using only as many stripes as there are files.

Note: NetBackup ignores the multistream directive for Instant Recovery backups.

With SQL Server backups performed with Snapshot Client, NetBackup identifies all of the backup streams by the same name, such as:

```
juneberry.MSSQL7.COLE.db.Northwind.~.7.001of003.20141012131132..C
```

and are differentiated by NetBackup by their backup IDs.

Restoring a multistreamed SQL Server backup with fewer devices than it was backed up with

In your recovery environment, you may have fewer drives available for restores than you used for backups. In this situation, SQL Server times out while it waits for the additional backup images to be mounted. To prevent this time out, modify the recovery batch file to specify the number of drives that are available for restore.

Consider, for example, if you had performed a backup using 5 drives, and only 2 are available for recovery. In the recovery batch file, change the stripes parameter from `STRIPES 5` to `STRIPES 2`. This change causes SQL Server to request two backup images at a time until all five images are restored.

Chapter 7

Protecting SQL Server data with VMware backups

This chapter includes the following topics:

About protecting SQL Server data with VMware backups

Through a VMware backup policy, NetBackup can create consistent full backups of an SQL Server database that resides on a virtual machine. To protect a supported application with a VMware policy, there is a new job or phase during the backup. An Application State Capture (ASC) job executes after the VMware discovery job and before the snapshot job(s). This ASC job contacts the NetBackup client on the

guest virtual machine. The ASC job collects and catalogs the specific data that is needed for application recovery.

You can do the following with VMware backups:

- Perform single pass VMware backups that can quiesce all instances of SQL Server in that guest OS and their databases.
- Use the existing SQL Server restore process to restore and recover data from VMware backups. From one VMware backup the following restore options are available: volume-level restore, file-level recovery, or database restore. You can also choose whether or not to truncate logs.
- Restore and recover databases from VMware backups to alternate clients. The target destination client can be a physical computer or a virtual machine.

About the Veritas VSS provider for vSphere

Veritas recommends that you use the Veritas VSS provider. To truncate logs, you must use the Veritas VSS provider to create full VSS backups. The VMware VSS provider creates copy-only backups, which cannot be used as a basis to truncate logs.

When the Veritas VSS provider is installed and NetBackup starts a virtual machine snapshot, VMware Tools calls the Veritas VSS provider to quiesce the VSS writers for a file-level consistent backup. If log truncation is enabled in the policy, the logs are truncated when the VMware snapshot is complete.

Note: The Veritas VSS provider must be installed separately.

See "Installing the Veritas VSS provider for vSphere" on page 105.

Limitations of using a VMware policy to protect SQL Server

The following limitations exist when you configure a VMware policy to protect SQL Server:

- This list is not a comprehensive list of VMware policy limitations. For additional information on support for NetBackup in virtual environments, see the following: http://www.veritas.com/docs/000006177
- VMware incremental backups of SQL Server are not supported with this version of NetBackup. However, the use of Accelerator may increase the speed of full backups.
- Point-in-time restores are not supported from VMware backups.
- SQL Servers cannot be clustered.

- VMware application backups are not support with availability groups (AGs).
- The Application State Capture (ASC) job fails and the databases are not protected if you do any of the following:
  - Disable the **Virtual Machine quiesce** option.
  - Exclude any data disks from the VMware policy, on the **Exclude Disks** tab. Be sure that any disks that you exclude do not contain SQL Server data.
- Databases are cataloged and protected only if they exist in a configuration that is supported for VMware backups. As long as there are any databases that can be protected, the ASC job continues. If you select databases for backup that exist on supported and on unsupported disks, the ASC job produces a status 1 (partially successful). The ASC job detects these situations and the job details include the result of the backup operation.

  SQL Server databases are not cataloged and backed up if they exist on the following:
  - Raw device mapping (RDMs). Make sure that the SQL Server virtual machine does not use RDM as storage for databases and transaction logs.
  - Virtual Machine Disk (vmdk) volumes that are marked as independent. Make sure that the SQL Server databases and transaction logs are not stored on independent disks.
  - Mount point volumes.
  - Virtual hard disks (VHDs).
    If NetBackup detects any database objects on a VHD disk, the ASC job fails and no SQL Server content is cataloged. All objects in the backup are not cataloged, including those that do not exist on the VHD.
  - RAID volumes.
  - ReFS file systems.
  - An excluded Windows boot disk. The ASC job detects this type of disk and treats it like an independent disk.
    The VMware backup cannot exclude for any reason the disk on which NetBackup is installed. For example, do not select the **Exclude boot disk** option if NetBackup is installed on the boot drive (typically C:).
- VMware policies let you exclude certain virtual disks from the VMware backup. If you want to exclude specific SQL Server components, use a MS-SQL-Server policy.

About configuring NetBackup for VMware backups that protect SQL Server

Table 7-1 Steps to configure VMware backups that protect SQL Server

Step	Action	Description
Step 1	Configure your VMware environment and NetBackup.	See the NetBackup for VMware Administrator's Guide. Each ESX server that hosts the database must have a license for NetBackup for SQL Server license and the Enterprise Client. Install the NetBackup client software on the virtual machines that have SQL Server running.
Step 2	Install the Veritas VSS provider.	The Veritas VSS provider creates full backups, which allows VMware backups to truncate logs. You can only truncate logs if there is an existing full backup. See "Installing the Veritas VSS provider for vSphere" on page 105.
Step 3	Configure the logon account for the NetBackup services.	The logon account for the NetBackup Client Service and the NetBackup Legacy Network Service must meet certain requirements. See "Configuring the NetBackup services for VMware backups that protect SQL Server" on page 106.
Step 4	(SQL Server 2012 and later) If you choose to truncate logs, ensure that the account that runs the Microsoft SQL Server Service has full permissions for the NetBackup Legacy Network Service `temp` directory.	This directory is `C:\Users\user\AppData\Local\Temp`. *User* is the account that runs the NetBackup Legacy Network Service.
Step 5	If you want to use Replication Director to manage your VMware snapshots and snapshot replicas, create a storage lifecycle policy (SLP).	This feature requires the NetBackup Replication Director license. See the NetBackup Replication Director Solutions Guide.

Table 7-1 Steps to configure VMware backups that protect SQL Server *(continued)*

Step	Action	Description
Step 6	Configure a VMware policy.	See "Configuring a VMware backup policy to protect SQL Server" on page 107. See "Configuring a VMware policy to protect SQL Server using Replication Director to manage snapshot replication" on page 109. See the NetBackup for VMware Administrator's Guide. Note that if you want to truncate logs, you must first perform a full backup without log truncation. See the following topic for more information. See "About truncating logs with a VMware backup that protects SQL Server" on page 111. Additional information is available on how to use Accelerator to potentially increase the speed of full VMware backups. See "Using NetBackup Accelerator to increase speed of full VMware backups" on page 104.
Step 7	If you use a **Primary VM identifier** other than **VM hostname**, you need to map that identifier to the host name of the VM.	Configure this mapping in the **Distributed Application Restore Mapping** host property on the master server. See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.
Step 8	Review the auto-discovered mappings for the hosts in your environment.	Approve each valid **Auto-Discovered Mapping** that NetBackup discovers in your environment. Perform this configuration in the **Host Management** properties on the master server. See "Reviewing the auto-discovered mappings in Host Management" on page 66.

Using NetBackup Accelerator to increase speed of full VMware backups

Select the **Use Accelerator** option to use NetBackup Accelerator to potentially increase the speed of full VMware backups. By reducing the backup time, it is easier

to perform the VMware backup within the backup window. To use this feature, you must first perform an initial backup with **Use Accelerator** enabled. Subsequent backup times can then be significantly reduced.

Accelerator support for SQL Server currently restricts backups to the full schedule type. This restriction also exists for a VMware backup that protects SQL Server without Accelerator.

See "Configuring a VMware backup policy to protect SQL Server" on page 107.

To periodically establish a new baseline of change detection on the client, create a separate policy schedule with the **Accelerator forced rescan** option enabled.

This feature requires an MSDP or PureDisk storage unit and the Data Protection Optimization Option license. For more details on Accelerator with VMware backups, see the NetBackup for VMware Administrator's Guide.

Installing the Veritas VSS provider for vSphere

To use the Veritas VSS provider you must install it manually following installation of the NetBackup for Windows client. If the VMware VSS provider is installed, the installation program removes it and may require a restart of the computer.

To install the Veritas VSS provider

1. Browse to the following location:

   install_path\Veritas\NetBackup\bin\goodies\

2. Double-click on the **Veritas VSS provider for vSphere** shortcut.
3. Follow the prompts.
4. When the utility has completed, restart the computer if prompted.
5. Following the restart, the utility resumes. Follow the prompts to complete the installation.

To uninstall the Veritas VSS provider

1. In the Control Panel, open **Add or Remove Programs** or **Programs and Features**.
2. Double-click on **Veritas VSS provider for vSphere**.

   The uninstall program does not automatically reinstall the VMware VSS provider.

Configuring the NetBackup services for VMware backups that protect SQL Server

NetBackup uses the NetBackup Client Service and the NetBackup Legacy Network Service to access the SQL Server when it performs VMware backups and restores. For VMware backups, the logon account must meet the following requirements:

- You cannot use the Local System account as the logon account.
- For VMware backups with Replication Director, the account has access to the CIFS shares on the NetApp disk array.
- The account has the fixed server role “sysadmin”. You can use a domain account, a member of BUILTIN\Administrators, or another account that has this role.
- Both services must use the same logon account.
- (SQL 2012 and later) If you choose to truncate logs, ensure that the account that runs the Microsoft SQL Server Service has full permissions for the NetBackup Legacy Network Service `temp` directory.
  This directory is `C:\Users\user\AppData\Local\Temp`. *User* is the account that runs the NetBackup Legacy Network Service.

Note: Configure the logon accounts for the services on the hosts that you use to browse for backups and the hosts you use to perform restores.

To configure the NetBackup services for VMware backups that protect SQL Server

1. Log on to the Windows host with the account that has the sysadmin role and any necessary local security privileges.
2. If the SQL Server host and instance use standard or mixed security, perform the following steps:
   - Open the NetBackup MS SQL Client.
   - Select **File > Set SQL Server connection properties**.
   - Provide the SQL Server **Userid** and **Password**.
   - Click **Apply**.
   - Click **Close**.
3. Open the Windows Services application.
4. Double-click the **NetBackup Client Service** entry.
5. Click on the **Log On** tab.

6 Provide the name of the logon account.

To change the logon account, you must have administrator group privileges.

The account must include the domain name, followed by the user account, ***domain_name**account***. For example, **recovery\netbackup**.

7 Click **OK**.

8 Double-click on the **NetBackup Legacy Network Service** entry.

9 Click on the **Log On** tab.

10 Provide the name of the logon account.

To change the logon account, you must have administrator group privileges.

The account must include the domain name, followed by the user account, ***domain_name**account***. For example, **recovery\netbackup**.

11 Click **OK**.

12 Stop and start the NetBackup Client Service and the NetBackup Legacy Network Service.

13 Close the Services control panel application.

Configuring a VMware backup policy to protect SQL Server

The following steps describe how to configure VMware backups of a SQL Server database.

To configure a VMware backup policy to protect SQL Server

1 Log on to the master server as administrator.

2 Start the NetBackup Administration Console.

3 Create a new policy or open the policy you want to configure.

4 In the **Policy** dialog box, click the **Attributes** tab.

5 From the **Policy type** list, select **VMware**.

6 In the **Policy storage** box, select a disk storage unit.

If you want to use NetBackup Accelerator, select a PureDisk storage unit type (MSDP or PureDisk). The NetBackup device mapping files list all supported storage types.

7 If you want to use NetBackup Accelerator, click **Use Accelerator**.

Accelerator uses the initial full backup to establish a baseline. Any subsequent backups that are performed with Accelerator can run significantly faster. You may want to create an additional policy schedule that enables the **Accelerator forced rescan** option. This option establishes a new baseline for the next Accelerator backup. For more details on NetBackup Accelerator, see the following:

See "Using NetBackup Accelerator to increase speed of full VMware backups" on page 104.

NetBackup for VMware Administrator's Guide

When you enable Accelerator, on the **VMware** tab the **Enable block-level incremental backup** option is also selected and grayed out.

8 On the **Schedules** tab, create a schedule for full backups.

9 On the **Clients** tab, click **Select automatically through query**.

10 Select **NetBackup host to perform automatic virtual machine selection** and the host you want to use.

11 Use the Query Builder to create a rule(s) that selects the virtual machines you want to back up.

For more details on the Query Builder, see the NetBackup for VMware Administrator's Guide.

12 Click the **Backup Selections** tab.

This tab displays the query you created on the **Clients** tab.

13 Click the **VMware** tab.

For details on the options in this dialog box, see the NetBackup for VMware Administrator's Guide.

14 Select the **Primary VM identifier** to use to catalog the backups.

15 Click **Enable file recovery from VM backup**.

This option must be enabled for application protection of SQL Server.

16 Click **Enable SQL Recovery**.

This option enables recovery of the SQL databases from the virtual machine backups. If this option is disabled, you can recover the entire virtual machine from the backup, but you cannot recover the databases individually.

17 If you want to truncate logs, do not enable **Truncate logs** at this time. You must first perform a full backup without log truncation, described later in this procedure.

18 If you want to exclude certain disks from the VMware backup, click the **Exclude Disks** tab.

NetBackup excludes those disks from the VMware backup that protects SQL Server. Be sure that any disks that you exclude do not contain Exchange data.

For details on the **Exclude Disks** tab, see the NetBackup for VMware Administrator's Guide.

19 Click **OK** to save the policy.

If you do not want to truncate transaction logs, no further action is necessary.

If you want to truncate transaction logs, continue with step 20.

20 Perform a full backup without log truncation.

Without this initial full backup, the ASC job fails.

21 When the backup completes, open the policy that you created in step 3.

22 Click the **VMware** tab.

23 Under **Enable SQL Server Recovery**, select **Truncate logs**.

For SQL Server, this option truncates the transaction logs when the VMware snapshot of the virtual machine is complete. For additional information on truncating logs and requirements, see the following topic.

See "About truncating logs with a VMware backup that protects SQL Server" on page 111.

24 Click **OK** to save the policy.

25 Perform a full VMware backup.

Configuring a VMware policy to protect SQL Server using Replication Director to manage snapshot replication

This topic describes how to configure a VMware policy to back up SQL Server using Replication Director to manage snapshot replication. Note that NetBackup must have access to the CIFS share on the NetApp disk array.

To configure a VMware policy to back up SQL Server using Replication Director to manage snapshot replication

1 Log on to the master server as administrator.

2 Start the NetBackup Administration Console.

3 Create a new policy or open the policy you want to configure.

4 In the **Policy** dialog box, click the **Attributes** tab.

5 From the **Policy type** list, select **VMware**.

6 In the **Policy storage** list select the storage lifecycle policy (SLP) that you want to use. This SLP must be configured for snapshot replication.

For complete details on how to configure Replication Director with VMware backups, see the NetBackup Replication Director Solutions Guide.

7 In the **Snapshot Client and Replication Director** group, click **Use Replication Director**.

8 On the **Schedules** tab, create a schedule for full backups.

9 On the **Clients** tab click **Select automatically through query**.

10 Select **NetBackup host to perform automatic virtual machine selection** and the host you want to use.

11 Use the Query Builder to create a rule(s) that select the virtual machines you want to back up.

For more details on the Query Builder, see NetBackup for VMware Administrator's Guide.

12 Click the **Backup Selections** tab.

This tab displays the query you created on the **Clients** tab.

13 Click the **VMware** tab.

For details on the options in this dialog box, see the NetBackup for VMware Administrator's Guide.

14 Select the **Primary VM identifier** to use to catalog the backups.

15 Click **Enable SQL Server Recovery**.

This option enables recovery of the SQL databases from the virtual machine backups. If this option is disabled, you can recover the entire virtual machine from the backup, but you cannot recover the databases individually.

16 If you want to truncate logs, do not enable **Truncate logs** at this time. You must first perform a full backup without log truncation, described later in this procedure.

17 Click **OK** to save the policy.

If you do not want to truncate transaction logs, no further action is necessary.

If you want to truncate transaction logs, continue with step 18.

18 Perform a full backup without log truncation.

19 When the backup completes, open the policy that you created in step 2.

20 Click the **VMware** tab.

21 Under **Enable SQL Server Recovery**, select **Truncate logs**.

For SQL Server, this option truncates the transaction logs when the VMware snapshot of the virtual machine is complete. For additional information on truncating logs and requirements, see the following topic.

See "About truncating logs with a VMware backup that protects SQL Server" on page 111.

22 Click **OK** to save the policy.

23 Perform a full VMware backup.

About truncating logs with a VMware backup that protects SQL Server

The following requirements exist if you want to truncate logs with a VMware backup that protects SQL Server:

- To create a full backup, you must install the Veritas VSS provider. The VMware VSS provider creates copy-only backups, which cannot be used as a basis to truncate logs.
  See "Installing the Veritas VSS provider for vSphere" on page 105.
- For SQL Server 2012 and later, the account that runs the Microsoft SQL Server Service must have full permissions for the NetBackup Legacy Network Service `temp` directory. This directory is `C:\Users\`*`user`*`\AppData\Local\Temp`. *User* is the account that runs the NetBackup Legacy Network Service.
- To truncate logs, you must first perform a full VMware backup without log truncation. Without this initial full backup, the ASC job fails. When this backup is complete, then enable log truncation in the policy.

Restoring SQL Server databases from a VMware backup

The following steps describe how to restore an SQL Server database from a full VMware backup.

To restore a SQL Server database from a VMware backup

1. Browse for the backup images you want to restore.

   See "Browsing for SQL Server backup images" on page 80.

2. In the **Restore Microsoft SQL Server Objects** dialog box, expand the database instance.
3. Expand the database.
4. Select the database image that you want to restore.

   Only the **Recovered** recovery option is available for VMware backups of SQL Server.

5. Click **Restore**.

Chapter 8

Using NetBackup for SQL Server with Snapshot Client

This chapter includes the following topics:

About NetBackup Snapshot Client for SQL Server

NetBackup for SQL Server includes support for snapshot backups. The snapshot technology uses SQL Server VDI (virtual device interface) quiescence to affect a momentary freeze on database activity. Then the agent can back up and restore SQL Server objects by taking snapshots of the component files. Data is captured at a particular instant. The resulting snapshot can be backed up without affecting the availability of the database. These snapshots are backed up to the storage unit.

A separate Snapshot Client license provides additional features for snapshot backups. You can configure the snapshot image for Instant Recovery and you can configure an alternate client to perform the snapshot backup.

The following NetBackup Snapshot Client features are available for use with NetBackup for SQL Server:

Snapshot backup	A point-in-time, read-only, disk-based copy of a client volume. NetBackup backs up data from the snapshot, not directly from the client's primary or original volume.
Instant Recovery	Makes the backups available for recovery from the local disk. The snapshot can also be the source for an additional backup copy to tape or other storage.
Off-host backup	Shifts the burden of backup processing onto a separate backup agent, reducing the backup impact on the client's computing resources. The backup agent sends the client's data to the storage device.

Although all of these features are provided through Snapshot Client support for SQL Server, not all snapshot methods are supported. For information on how to select a method, see the NetBackup Snapshot Client Administrator's Guide. For a description of snapshot methods available for use with NetBackup for SQL Server, see the NetBackup Snapshot Client compatibility list.

How SQL Server operations use Snapshot Client

This section describes how SQL Server operations use the Snapshot Client.

About selection of backup method

The selection of a backup methodology, whether standard or Snapshot Client, is dependent on what policy is used. If a policy configured for Snapshot Client is selected, then additional attributes of policy determine the Snapshot Client features. It also determines the specific snapshot methods that are used.

About SQL Server limitations with snapshots

Due to SQL Server limitations certain objects cannot be backed up by snapshots. These are database differentials, filegroup differentials, and transaction logs. If a Snapshot Client policy is selected to back up one of these object types, then NetBackup performs a stream-based backup. NetBackup uses the storage unit that is provided in the policy configuration. If a storage unit is not provided, then NetBackup uses the default storage unit for the server.

What is backed up by NetBackup for SQL Server

The database administrator works exclusively with logical objects, such as databases and filegroups. However, it is useful to understand the differences between file- and stream-based backups in terms of the data content that is archived. For stream-based backups, NetBackup captures the data stream content that is provided by SQL Server. If the user has specified multiple streams, then SQL Server opens multiple streams that NetBackup catalogs as separate images.

For file-based backups, NetBackup creates a file list that consists of all the physical files that constitute the object. This file list is supplied to the Snapshot Client, which is responsible for snapshot creation. If multiple streams are specified, then NetBackup divides the file list into sub-lists. Each sub-list is backed up separately and constitutes a separate image. Users may notice that if multiple streams are specified for a file-based backup and if the number of streams exceeds the number of component files, then the number of file-based streams does not exceed the number of files. With stream-based SQL Server backups, SQL Server always creates exactly the number of streams that the end user specifies.

The file list that is used to back up a SQL Server database consists of the physical files that constitute the primary filegroup. The file list also consists of any secondary filegroups, and the transaction log. Typically, these can be identified respectively by their name extensions, which are `.mdf`, `.ndf`, and `.ldf`. The file list for a filegroup backup consists of the physical files that belong to the filegroup. And, finally, the file list for a file object backup consists of a single physical file. This file is the file that maps to the SQL Server file object.

About Snapshot Client and SQL Server performance considerations

When a physical file is backed up with the Snapshot Client, the backup consists of the entire extent. This backup contrasts with stream-based SQL Server backups where only the actual data content of the objects are archived. If you intend to use snapshot technology to back up SQL Server, you may want to use the SQL Server dynamic file allocation. This configuration reduces the likelihood that any of the component files contain large areas of empty space.

Also review the other considerations for SQL Server disk initialization.

See “About NetBackup for SQL performance factors” on page 70.

About SQL Server snapshot backups

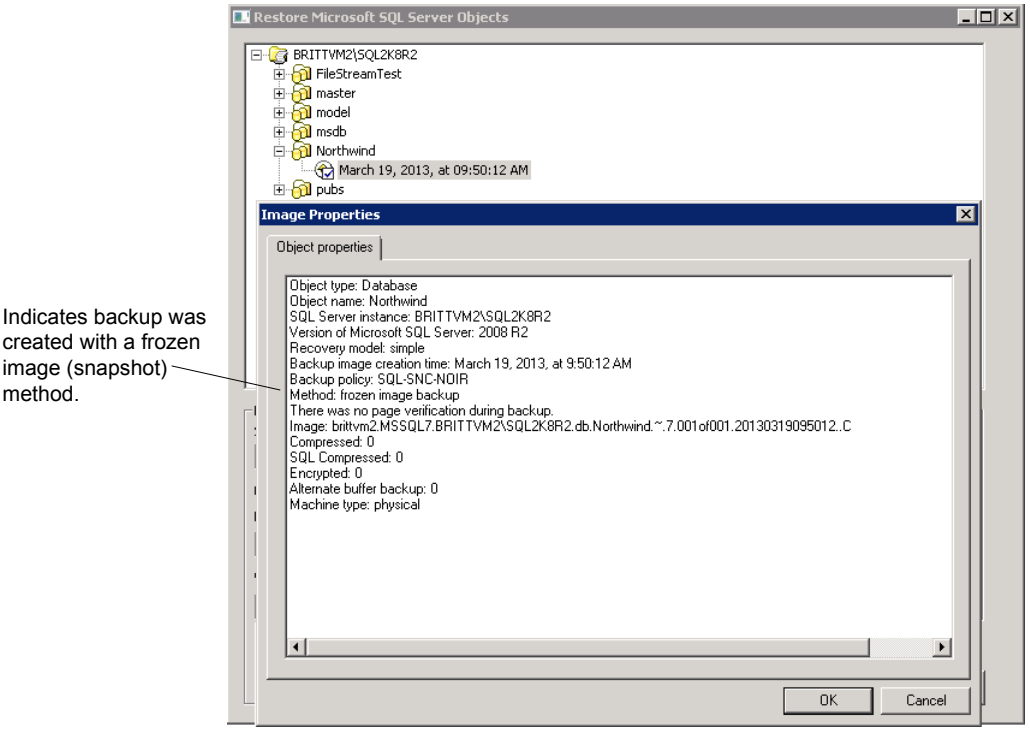
No special interfacing considerations exist when you perform Snapshot Client backups of SQL Server. A snapshot backup is performed if the backup object is: a database, a filegroup, or a file and a policy is selected and configured for Snapshot Client. If a differential backup or transaction log backup is tried with a Snapshot

Client backup, then the operation uses the selected policy. But a standard database backup is performed with the configured storage unit.

About SQL Server snapshot restores

Any backup images that were created from snapshots display along with standard backup images. That is, all backup items—without regard to method—display in a time-sequenced ordering that respects the composition of the database hierarchy. In addition, no weighting is given in to determine an optimal recovery that is based on the backup method. To determine what backup method and policy were used when a SQL Server backup was created, right-click the backup image and select **Properties**.

Figure 8-1 Backup method that appears in the backup image properties



Configuration requirements for SQL Server snapshot and Instant Recovery backups

Review the following requirements before you configure NetBackup for SQL Server with snapshot backups:

- See the NetBackup Snapshot Client Administrator's Guide for details on the hardware requirements and software requirements for the snapshot method that you want to use.
- Go to the Veritas Support website for details on the snapshot methods and platforms that are supported for NetBackup for SQL Server.
- The volume(s) which contains the SQL Server databases and log files should be dedicated to SQL Server only. Other types of databases (e.g., Exchange) should not reside on the volume(s).
- NetBackup Snapshot Client is installed and configured correctly and you have a the license for this option. See the NetBackup Snapshot Client Administrator's Guide for details.
- Only one snapshot method can be configured per policy. If you want to use a different snapshot method different clients, then create a separate policy for each group of clients and the snapshot method you want to use. Then select one method for each policy.

Configuring a snapshot policy for SQL Server

These instructions describe how to configure a Snapshot Client policy. Optionally you can choose to perform an off-host backup. This topic only covers what is necessary to configure snapshot backups for a MS-SQL-Server policy.

See "About SQL Server Intelligent Policies" on page 43.

See "Adding a new SQL Server legacy policy" on page 201.

To configure a snapshot policy for SQL Server

1. For SQL Server legacy policies, create a backup script (`.bch` file) using the NetBackup MS SQL Client.
2. Open the policy you want to configure.
3. Click the **Attributes** tab.
4. From the **Policy type** list, select **MS-SQL-Server**.

5 Select the **Policy storage**.

If database differentials, filegroup differentials, or transaction logs are included in the **Backup Selections** list of a policy that uses Snapshot Client, then NetBackup performs a stream-based backup. The selected storage unit is used. If a storage unit is not provided, then NetBackup uses the default storage unit for the server.

6 Select **Perform snapshot backups**.

7 Choose to have NetBackup select the snapshot method or select the snapshot method manually.

Perform one of the following:

- By default, NetBackup chooses a snapshot method for you. If you have changed this setting and want NetBackup to choose the method automatically, click **Snapshot Client Options**. Then from the **Snapshot method** list, select **auto**.
- To use a specific snapshot method, click **Snapshot Client Options**. From the **Snapshot method** list, select the method you want to use for this policy.

See the NetBackup Snapshot Client Administrator's Guide for details on how to select the snapshot method and automatic snapshot selection.

8 (Optional) To use an alternate client to reduce the processing load on the client, perform the following steps:

- The alternate client must be the client that shares the disk array. This option may require additional configuration. See the NetBackup Snapshot Client Administrator's Guide.
- Select **Perform off-host backup**.
- Click **Use alternate client** and enter the name of the alternate client.

Note: Use data mover is not a supported option for NetBackup for SQL Server.

---

9 On the **Instances and Databases** tab, choose how you want to protect SQL Server:

- (SQL Server Intelligent Policy) Choose **Protect Instances** or **Protect instance groups**.
  If you choose the instances option, you can select either individual instances or databases.
  See "Adding instances to a policy" on page 50.
  See "Adding databases to a policy" on page 51.
  See "Adding instance groups to a backup policy" on page 57.

- (SQL Server legacy policies) Choose **Clients for use with batch files**.

10 (SQL Server Intelligent Policy) Add other policy information as follows:

- Add schedules.
  See "About schedule properties" on page 46.
- (Optional) Select specific filegroups or files that you want to back up. By default, NetBackup backs up an entire database.
  See "Adding filegroups or files to the backup selections list" on page 55.
- (Optional) Make changes to any tuning parameters.
  See "About tuning parameters for SQL Server backups" on page 58.

11 (SQL Server legacy policies) Add other policy information as follows:

- Add schedules.
  See "About schedule properties " on page 202.
- Add clients.
  See "Adding clients to a policy" on page 207.
- Add batch files to the backup selections list.
  See "Adding batch files to the backup selections list " on page 208.

12 Click **OK** to save the policy.

Configuring a policy for Instant Recovery backups of SQL Server

These instructions describe how to configure a policy for Instant Recovery. Optionally you can choose to back up to disk only. This topic only covers what is necessary to configure Instant Recovery backups for a MS-SQL-Server policy.

See "About SQL Server Intelligent Policies" on page 43.

See "Adding a new SQL Server legacy policy" on page 201.

To configure a policy for Instant Recovery

1 For SQL Server legacy policies, create a backup script using the NetBackup MS SQL Client interface.

2 Open the policy you want to configure.

3 Click the **Attributes** tab.

4 From the **Policy type** list, select **MS-SQL-Server**.

5 Select the **Policy storage**.

If you select an Instant Recovery option on the **Schedules** tab (see step 10), the storage unit is not used. NetBackup creates only a disk snapshot.

If database differentials, filegroup differentials, or transaction logs are included in the policy, then NetBackup performs a stream-based backup. This backup uses the selected storage unit. If a storage unit is not provided, then NetBackup uses the default storage unit for the server.

6 Click **Perform snapshot backups**.

7 Choose to have NetBackup select the snapshot method or select the snapshot method manually.

Perform one of the following:

- By default, NetBackup chooses a snapshot method for you. If you have changed this setting and want NetBackup to choose the method automatically, click **Snapshot Client Options**. In the **Snapshot Client Options** dialog box, from the **Snapshot method** list, choose **auto**.
- To use a specific snapshot method, click **Snapshot Client Options**. In the **Snapshot Client Options** dialog box, from the **Snapshot method** list, choose the method you want to use for this policy.

See the NetBackup Snapshot Client Administrator's Guide for details on how to select the snapshot method and automatic snapshot selection.

8 Select **Retain snapshots for Instant Recovery**.

NetBackup retains the snapshot on disk, so that Instant Recovery can be performed from the snapshot.

A normal backup to storage is also performed, if you do not choose to create a snapshot only (see step 10).

9 On the **Instances and Databases** tab, choose how you want to protect SQL Server:

- (SQL Server Intelligent Policy) Choose **Protect Instances** or **Protect instance groups**.
  If you choose the instances option, you can select either individual instances or databases.
  See "Adding instances to a policy" on page 50.
  See "Adding databases to a policy" on page 51.
  See "Adding instance groups to a backup policy" on page 57.
- (SQL Server legacy policies) Choose **Clients for use with batch files**.

10 To configure schedules, click the **Schedules** tab.

- (SQL Server Intelligent Policies) Configure a full backup schedule.
  See "About schedule properties" on page 46.
- (Legacy policies) Follow the instructions to configure an Application and a full backup schedule.
  See "About schedule properties " on page 202.

11 (Optional) To create a disk image only, open the Full Backup schedule (Intelligent Policies) or the Application schedule (legacy policies) and select an Instant Recovery option.

Select one of the following options:

- If **Snapshots and copy snapshots to a storage unit** is selected, NetBackup creates a disk snapshot. NetBackup also backs up the client's data to the storage unit that is specified for the policy.
- If **Snapshots only** is selected, the image is not backed up to tape or to other storage. NetBackup creates a disk snapshot only. Note that this disk snapshot is not considered a replacement for traditional backup.

12 (SQL Server Intelligent Policy) Add other policy information as follows:

- (Optional) Select specific filegroups or files that you want to back up. By default, NetBackup backs up an entire database.
  See "Adding filegroups or files to the backup selections list" on page 55.
- (Optional) Make changes to any tuning parameters.
  See "About tuning parameters for SQL Server backups" on page 58.

13 (SQL Server legacy policies) Add other policy information as follows:

- Add clients.
  See "Adding clients to a policy" on page 207.
- Add batch files to the backup selections list.
  See "Adding batch files to the backup selections list " on page 208.

14 Click **OK** to save the policy.

Using copy-only snapshot backups to affect how differentials are based

SQL Server records the history of successful database backups in the msdb system database. It uses this history in to decide how to base differential backups. In particular, SQL Server creates differential database backups as cumulative with respect to the last full database backup that it has recorded in the msdb. This action

allows for a quick recovery in case a failure has been detected after the last full database backup.

Assume that full backups are created every day at midnight; differentials are created every day at 6AM, noon, and 6PM; and transaction log backups are created every two hours. If a failure occurs at 7:50 P.M. on Tuesday, then a point in time recovery could be achieved with a restore of: the full database from Tuesday at 12AM, followed by the differential at 6PM on Tuesday, and finally the transaction log at 8PM (choose "to 7:50 P.M.").

However, with Instant Recovery backups, you may not retain the daily full backup after the next full backup is created. If you require a point in time restore before the latest backup, the differentials are based on the backups that no longer exist. The alternative is to recover based on the last full backup that was retained. And you have to use a potentially long sequence of transaction log images.

To resolve this issue, NetBackup lets you create the SQL Server Snapshot Client backups that are not recorded in the msdb. To create these backups, NetBackup uses the copy-only backup feature, which allows the backups to be created as out-of-band.

Sample backup schedule using copy-only backups

To understand how recovery staging works with copy-only, consider a sample backup schedule with the following characteristics:

- The transaction log is backed up frequently, e.g., every two hours
- A full backup is saved to secondary tape storage once every several days
- Differential database backups are created several times per day
- An Instant Recovery backup is created several times per day and expires when the next one is created. This backup is created as copy-only.

Table 8-1 shows an excerpt from this schedule.

Table 8-1 Sample backup schedule using copy-only backups

Time	A full backup saved to secondary storage	Differential backup	PFI Copy-Only	Transaction log backup
Day 1				
12:00 A.M.	X			X
2:00 A.M.				X

Table 8-1 Sample backup schedule using copy-only backups *(continued)*

Time	A full backup saved to secondary storage	Differential backup	PFI Copy-Only	Transaction log backup
4:00 A.M.		X		X
6:00 A.M.			X	X
8:00 A.M.				X
10:00 A.M.		X		X
12:00 P.M.			X	X
2:00 P.M.				X
4:00 P.M.		X		X
6:00 P.M.			X	X
8:00 P.M.				X
10:00 P.M.		X		X
Day 2				
12:00 A.M.			X	X
2:00 A.M.				X

Under this schedule, full backups are performed every six hours. If a failure occurs, and is detected immediately, then you can restore the last full backup. Then you can replay, on average, three hours of transaction logs to achieve recovery. However, if a failure is not detected until after the next full backup, then there are not any full backups available. There are none available since 12:00 A.M. on day 1. The Instant Recovery backups are copy-only. However, the differential backups would each be cumulative with respect to the last full backup that is not copy-only.

In this example, suppose that an error occurs at 11:30 P.M. on day 1. But the error is not detected until 12:30 A.M. on day 2, after the 12:00 A.M. full backup. Since the 6:00 P.M. full backup no longer exists it would be necessary to begin the recovery with the backup taken at 12:00 A.M. on day 1. However, since all of the full backups were copy-only since then, the differential backup from 10:00 P.M. would be cumulative with respect to that backup. The recovery sequence would be restore the 12:00 A.M. day 1 backup. Restore the 10:00 P.M. differential backup. Restore the 1½ hours of transaction log backups.

The copy-only attribute appears in the properties for the snapshot backup image. Differential backups are automatically associated with the correct full backup. The SQL Agent recognizes these backups when it selects the recovery set for the full database restore.

Creating a copy-only backup (legacy SQL Server policies)

Any backup can be created as copy-only. An Instant Recovery backup is automatically created as copy-only. For legacy SQL Server policies, set the `COPYONLY TRUE` setting in the backup batch file. For SQL Server Intelligent Policies, enable **Copy-only backup** on the **Microsoft SQL Server** tab.

See "About tuning parameters for SQL Server backups" on page 58.

To create a copy-only backup

1. Open an existing batch file in a text editor.
2. Insert the following:

```
COPYONLY TRUE
```

3. Save the batch file.

Creating an Instant Recovery backup that is not copy-only (legacy SQL Server policies)

For Instant Recovery backups, NetBackup automatically creates the backup image as copy-only. You can choose *not* to create the backup as copy-only.

To create an Instant Recovery backup that is not copy-only

1. Open an existing batch file in a text editor.
2. Insert the following:

```
COPYONLY FALSE
```

3. Save the batch file.

About SQL Server agent grouped backups (legacy SQL Server policies)

Note: This feature is only available with legacy SQL Server backup policies.

The SQL Server agent provides a method in which multiple databases can be quiesced together and split-off to form a single snapshot. This method minimizes the usage of system resources if the databases exist on a single volume. This happens because the aggregation of constituent files uses one snapshot volume instead of one per database. The method for aggregating database Snapshot Client backups is called backup "grouping".

When databases are backed up in a group, all of the databases are quiesced simultaneously. The constituent files of all databases are backed up to a single storage image under the same backup ID. This means that an "import and copy" procedure would use only one image to export all of the database backups in the group.

Requirements for a grouped backup

Certain requirements must be met for a grouped backup to be performed. If any of the following requirements are not met, a standard backup is performed:

- All backup operations must be full backups. Differential backups are not supported.
- The master database cannot be included in a grouped backup.
- The same policy must be specified for each backup operation in the group.
- The same NetBackup server must be specified for each backup operation in the group.

The simplest way to use grouped backup is to select multiple databases using the Backup Microsoft SQL Server Objects dialog box. If the conditions described apply, then the selected databases are backed up as a group.

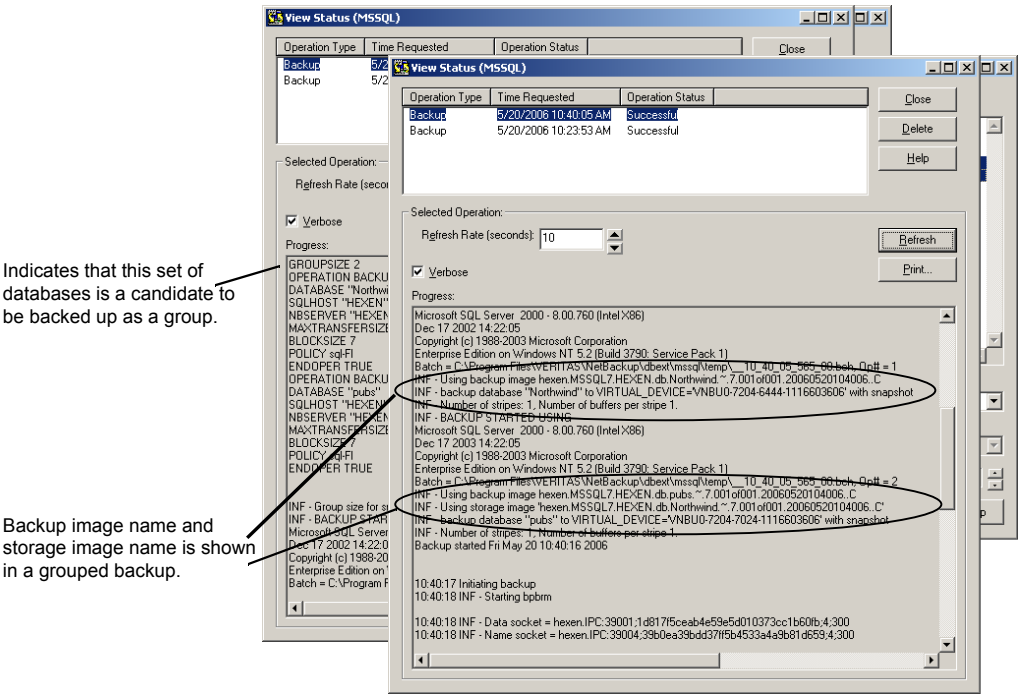
Viewing the progress of a grouped backup

You can determine that a grouped backup is underway from the progress report.

See Figure 8-2.

The keyword GROUPSIZE appears at the beginning of the batch file. This keyword indicates that NetBackup uses grouping to back up the selected SQL Server databases. If the appropriate conditions apply all operations are full database backups. Then all of the databases are snapped and backed up as a group. When this action happens, the progress log displays the backup image name as well as the storage image for each database in the group.

Figure 8-2 Progress report for a grouped backup operation



Restoring a database backed up in a group

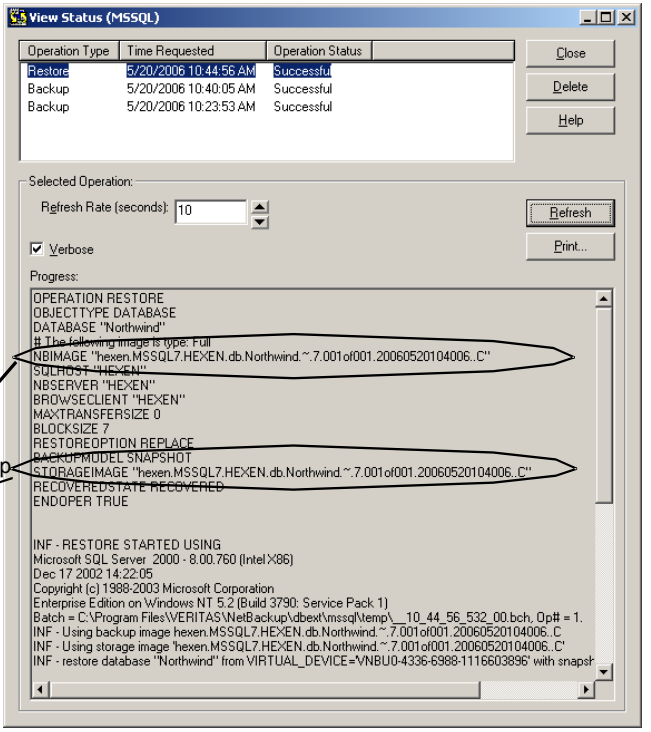
A database that is backed up in a group can be restored like any other database.

See "Restoring a SQL Server database backup" on page 84.

When you launch the restore operation, note that the batch file specifies the storage image name and the backup image name.

See Figure 8-3 on page 127.

Figure 8-3 Batch file shown in the progress report for the restore operation



Storage image name and backup image name are shown when restoring from a grouped backup.

Chapter 9

Protecting SQL Server in high availability (HA) environments

This chapter includes the following topics:

About SQL Server high availability (HA) environments

SQL Server Intelligent policies support the following types of SQL Server HA environments: SQL Server clusters and log-shipping. For log-shipping, the same caveats as for legacy SQL Server policies.

SQL Server legacy policies support the following types of SQL Server HA environments: AlwaysOn Availability Groups™, SQL Server clusters, database mirroring, and log-shipping.

For complete descriptions of each of these HA solutions refer to *SQL Server Books Online*. Each solution synchronizes one or more copies of selected databases on alternate SQL Server installations. A manual or an automatic failover results in continued access for mission critical database applications.

See "About using NetBackup to protect SQL Server availability groups" on page 129.

See "Configuring backups of clustered SQL Server instances (SQL Server Intelligent Policy) " on page 147.

See "Configuring backups of clustered SQL Server instances (legacy SQL Server policies)" on page 149.

See "About NetBackup for SQL Server with database mirroring" on page 150.

See "Configuring NetBackup to support database log-shipping" on page 153.

About using NetBackup to protect SQL Server availability groups

NetBackup for SQL Server supports backups and restores of SQL Server AlwaysOn Availability Group databases. For information on supported versions and environments, see the Application/Database Agent Compatibility List.

Note the following when you configure and run backup policies for availability groups (AGs):

- NetBackup supports backups of AGs with legacy backup policies (with the option Clients for use with batch files). Intelligent Policies do not support AG backups at this time.
- You can protect an AG environment in the following ways:
  - With a policy that protects the preferred replica.
    See "About protecting the preferred replica in a SQL Server availability group (legacy backup policies)" on page 130.
  - With a policy that protects a specific node in the AG.
    See "About protecting a specific node in a SQL Server availability group " on page 136.
- See "Configuring SQL Server backups when an availability group crosses NetBackup domains" on page 140.
  See "About protecting a specific node in a SQL Server availability group " on page 136.

- NetBackup supports backups of AGs in multi-NIC environments. For more information, see the following topic:
  See "About configuration of SQL Server backups with multiple NICs" on page 171.
- The timestamp for AG backup images reflects Coordinated Universal Time (UTC).

Limitations of backups of availability groups

Note the following limitations for backups of availability groups (AGs):

- NetBackup does not support the following types of backups for AG databases:
  - Snapshot backups of filegroups or files
  - Instant Recovery backups
  - VMware backups
  - Backups of non-readable secondary replicas
    NetBackup can only back up databases in a replica when you allow user connections for the replica.
    If a secondary replica is the preferred replica and it is non-readable, the backup fails. If a secondary replica is not the preferred replica, NetBackup skips the backup of that replica.

SQL Server does not support the following types of backups on a secondary replica:

- Full backups
  If a full backup takes place on a secondary replica, NetBackup converts the full backup to a copy-only backup.
- Differential backups
  Backups of this type result in a failed backup.
- Copy-only transaction log backups
  Backups of this type result in a failed backup.

About protecting the preferred replica in a SQL Server availability group (legacy backup policies)

This topic describes how to protect the preferred replica in a SQL Server availability group (AG).

Note the following when you configure a NetBackup policy to protect the preferred replica:

- In the backup batch file that you create use the `PREFERREDREPLICA TRUE` keyword, which honors your SQL Server backup preferences. These preferences

include the preferred replica, backup priority, and excluded replicas. Include this keyword in each backup operation in the batch file.

- NetBackup backs up the preferred replica, as determined by SQL Server. NetBackup can only fully protect the AG environment if the backup policy includes each node in the AG in the **Clients** list. Also, all batch files in the **Backup Selections** list must exist on all the AG nodes.
- Review the information on support and limitations for AGs.
  See "About using NetBackup to protect SQL Server availability groups" on page 129.

Note: Perform the following configuration steps after you create the SQL Server availability group.

Table 9-1 About protecting the preferred replica in a SQL Server availability group

Step	Action	Description
Step 1	Verify that you have a supported SQL Server configuration.	See the Application/Database Agent Compatibility List.
Step 2	On each node in the availability group, install the NetBackup client.	See "NetBackup server and client requirements" on page 21.
Step 3	On each node in the availability group, configure the NetBackup services.	See "Configuring the NetBackup services for SQL Server backups and restores " on page 186.
Step 4	Configure the mappings for distributed application restores.	Map the WSFC (Windows Server Failover Cluster) name to each AG node. If you have an AG with a FCI, you must configure additional mappings. Configure these mappings in the **Distributed Application Restore Mapping** host property on the master server. See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.
Step 5	Review the auto-discovered mappings for the hosts in your environment.	Approve each valid **Auto-Discovered Mapping** that NetBackup discovers in your environment. Perform this configuration in the **Host Management** properties on the master server. See "Reviewing the auto-discovered mappings in Host Management" on page 66.

Table 9-1 About protecting the preferred replica in a SQL Server availability group *(continued)*

Step	Action	Description
Step 6	Create a policy for each type of backup you want to perform.	See "Configuring an automatic backup policy for preferred replica of a SQL Server availability group (legacy SQL Server policies)" on page 132.
Step 7	On each node in the availability group, create a batch file for each type of backup that you want to perform.	See "Creating batch files for the policy that protects the preferred replica" on page 134. See "About using batch files with NetBackup for SQL Server" on page 188.
Step 8	Add the batch files to the policies that you created.	See "Adding the batch files to the policy that protects the preferred replica" on page 135.

Configuring an automatic backup policy for preferred replica of a SQL Server availability group (legacy SQL Server policies)

This topic describes how to create a backup policy for automatic (scheduled) backups of the preferred replica in a SQL Server availability group (AG). Create a policy for each type of backup that you want to perform. For example:

Policy A	Schedules: Full backup, run weekly
	Backup Selections: Batch file for full backups
	Clients: Node A, Node B, Node C
Policy B	Schedules: Full backup, run daily
	Backup Selections: Batch file for full differential backups
	Clients: Node A, Node B, Node C
Policy C	Schedules: Full backup, run per your RTO and RPO
	Backup Selections: Batch file for transaction log backups
	Clients: Node A, Node B, Node C

To configure an automatic backup policy for the preferred replica of a SQL Server availability group

1. Log on to the master server as administrator (Windows) or root (UNIX).
2. Open the NetBackup Administration Console.

3 If your site has more than one master server, choose the one on which you want to add the policy.

4 Select **Actions > New > Policy**.

5 In the **Add a New Policy** dialog box, in the **Policy name** box, type a unique name for the new policy.

6 Click **OK**.

7 On the **Attributes** tab, configure the following:

- Select the **MS-SQL-Server** policy type.
- Specify a storage unit.

See "About policy attributes" on page 45.

8 On the **Instances and Databases** tab, select **Clients for use with batch files**.

The tab name changes to **Clients** and the **Backup Selections** tab now lets you specify and browse for scripts.

9 On the **Schedules** tab, add a **Full Backup** schedule.

NetBackup also creates a Default-Application-Backup schedule. Use this schedule to set the retention level for the policy. See the NetBackup Administrator's Guide for more information.

See "About schedule properties " on page 202.

10 On the **Clients** tab, add the name of each node in the availability group.

Use the NetBackup client name for each node. If a replica is hosted on a failover cluster instance (FCI), use the virtual cluster instance name.

11 Click **OK** to save the policy.

12 Repeat step 4 to 11 in this procedure to create a policy for each type of backup (full, full differential, transaction log) that you want to perform.

Each type of backup requires a separate policy.

13 On each node in the AG, create a batch file for each type of backup that you want to perform with each policy.

See "Creating batch files for the policy that protects the preferred replica" on page 134.

Creating batch files for the policy that protects the preferred replica

This topic describes how to create batch files for the backup policies that protect the availability group (AG). These batch files also use the PREFERREDREPLICA keyword so that NetBackup protects the SQL Server preferred replica.

To create the batch files for an AG, you must log on to each node separately. Then use the NetBackup MS SQL Client to create the batch files on each node.

To create batch files for the policy that protects the preferred replica

1. This procedure assumes that you already created a separate policy for each type of backup that you want to perform.

   See "Configuring an automatic backup policy for preferred replica of a SQL Server availability group (legacy SQL Server policies)" on page 132.

2. Perform steps 3 to 14 in this procedure on each node in the AG.

   You must log on to each node separately and create the batch files from that node. This way the batch files have the correct settings for each node. Backups may fail if you create a batch file on one node and copy it to the other nodes in the AG.

3. Log on to one of the nodes in the AG.
4. Open the NetBackup MS SQL Client.
5. Select **File > Set SQL Server connection properties**.
6. From the **Instance** drop-down list, select the instance that hosts the AG.
7. Select **File** > **Backup SQL Server objects**.
8. Select the objects you want to backup in one of the following ways:
   - Select one or more databases, filegroups, or files.
   - To back up all databases, including the system databases (`DATABASE $ALL`), select the instance. From the **Back up** group, select **All**.
9. Select the **Type of Backup** and any other settings.
10. In the **NetBackup Policy** field, enter the name of the MS-SQL Server policy that you created.
11. From the **Backup script** group, select **Save**.
12. Click **Backup** and open the batch file.
13. For each operation in the batch file, add the keyword `PREFERREDREPLICA TRUE`.

    This keyword instructs NetBackup to perform the backup only on the preferred replica and not all the nodes in the AG.

14 Save and close the batch file.

Note the location of the batch file. Save the batch file for each node to the same file location. This way you only need to enter one file location for the batch file in the **Backup Selections** list.

15 Repeat steps 7 to 14 for any other types of backups that you want to perform. For example, full, full differential, or transaction log.

More information is available on how to create batch files.

See "About using batch files with NetBackup for SQL Server" on page 188.

16 Repeat the steps in this procedure (steps 3 to 15) to create batch files for the other AG nodes.

17 When you have created batch files for all the nodes in the AG, add the batch files to the policies that you created previously.

See "Adding the batch files to the policy that protects the preferred replica" on page 135.

Adding the batch files to the policy that protects the preferred replica

This topic describes how to add the batch files that you created to the backup policy that protects the preferred replica in the availability group (AG).

To add the batch files to the policy that protects the preferred replica

1 This procedure assumes that you already created a policy and created batch files on each node in the AG.

See "Configuring an automatic backup policy for preferred replica of a SQL Server availability group (legacy SQL Server policies)" on page 132.

See "Adding the batch files to the policy that protects the preferred replica" on page 135.

2 Open the policy that you created.

3 On the **Backup Selections** tab, add the batch files that you created for each AG node.

Include batch files for only one type of backup in this policy. (For example, full, full differential, or transaction log.)

4 Click **OK** to save the policy.

5 Repeat the steps in this procedure for each policy that you created.

About protecting a specific node in a SQL Server availability group

This topic describes how to protect a specific node in a SQL Server availability group (AG) using a legacy SQL Server policy.

Note the following when you configure a NetBackup policy to protect a specific node in an availability group:

- For this backup scenario, do not use the `PREFERREDREPLICA TRUE` keyword in your batch files. Backups are skipped if the backup policy does not include the node that hosts the preferred replica.
- Review the information on support and limitations for AGs.
  See "About using NetBackup to protect SQL Server availability groups" on page 129.

Note: Perform the following configuration steps after you create the SQL Server availability group.

Table 9-2 About protecting a specific node in a SQL Server availability group

Step	Action	Description
Step 1	Verify that you have a supported SQL Server configuration.	See the Application/Database Agent Compatibility List.
Step 2	On the node that you want to protect, install the NetBackup client.	See "NetBackup server and client requirements" on page 21.
Step 3	On the node that you want to protect, configure the NetBackup services.	See "Configuring the NetBackup services for SQL Server backups and restores " on page 186.
Step 4	Create a policy for each type of backup you want to perform.	See "Configuring an automatic backup policy for a specific node of a SQL Server availability group" on page 137.
Step 5	On the node that you want to protect, create a batch file for each type of backup that you want to perform.	See "Creating a batch file for the policy that protects a specific node in an availability group" on page 138. See "About using batch files with NetBackup for SQL Server" on page 188.
Step 6	Add the batch files to the policies that you created.	See "Adding the batch files to the policy that protects a specific node in the availability group" on page 139.

Configuring an automatic backup policy for a specific node of a SQL Server availability group

This topic describes how to create a backup policy for automatic (scheduled) backups of a specific node in a SQL Server availability group (AG). Create a policy for each type of backup that you want to perform. For example:

Policy A	Schedules: Full backup, run weekly
	Backup Selections: Batch file for full backups
	Clients: Node A
Policy B	Schedules: Full backup, run daily
	Backup Selections: Batch file for full differential backups
	Clients: Node A
Policy C	Schedules: Full backup, run per your RTO and RPO
	Backup Selections: Batch file for transaction log backups
	Clients: Node A

To configure an automatic backup policy for a specific node of a SQL Server availability group

1. Log on to the master server as administrator (Windows) or root (UNIX).
2. Open the NetBackup Administration Console.
3. If your site has more than one master server, choose the one on which you want to add the policy.
4. Select **Actions > New > Policy**.
5. In the **Add a New Policy** dialog box, in the **Policy name** box, type a unique name for the new policy.
6. Click **OK**.
7. On the **Attributes** tab, configure the following:
   - Select the **MS-SQL-Server** policy type.
   - Specify a storage unit.

   See "About policy attributes" on page 45.
8. On the **Instances and Databases** tab, select **Clients for use with batch files**.

   The tab name changes to **Clients** and the **Backup Selections** tab now lets you specify and browse for scripts.

9 On the **Schedules** tab, add a **Full Backup** schedule.

NetBackup also creates a Default-Application-Backup schedule. Use this schedule to set the retention level for the policy. See the NetBackup Administrator's Guide for more information.

See "About schedule properties " on page 202.

10 On the **Clients** tab, add the name of the node in the AG that hosts the replica that you want to protect.

Use the NetBackup client name for the node. If a replica is hosted on a failover cluster instance (FCI), use the virtual cluster instance name.

11 Click **OK** to save the policy.

12 Repeat the step 4 through step 11 in this procedure to create a policy for each type of backup (full, full differential, transaction log) that you want to perform.

Each type of backup requires a separate policy.

13 Create a batch file for each type of backup that you want to perform with each policy.

See "Creating a batch file for the policy that protects a specific node in an availability group" on page 138.

Creating a batch file for the policy that protects a specific node in an availability group

This topic describes how to create batch files for the backup policies that protect a specific node in the availability group (AG).

To create batch files for the policy that protects a specific availability group node

1 This procedure assumes that you already created a policy.

See "Configuring an automatic backup policy for a specific node of a SQL Server availability group" on page 137.

2 Log on to the AG node that hosts the replica you want to protect.

3 Open the NetBackup MS SQL Client.

4 Select **File > Set SQL Server connection properties**.

5 From the **Instance** drop-down list, select the instance that hosts the AG.

6 Select **File** > **Backup SQL Server objects**.

7 Select the objects you want to backup in one of the following ways:

- Select one or more databases, filegroups, or files.

- To back up all databases, including the system databases (`DATABASE $ALL`), select the instance. From the **Back up** group, select **All**.

8 Select the **Type of Backup** and any other settings.

9 In the **NetBackup Policy** field, enter the name of the MS-SQL Server policy that you created.

10 From the **Backup script** group, select **Save**.

11 Click **Backup** and save the batch file.

Do not use the `PREFERREDREPLICA TRUE` keyword in your batch files. Backups are skipped if the backup policy does not include the node that hosts the preferred replica.

12 Repeat steps 6 to 11 for any other the types of backups that you want to perform. For example, full, full differential, or transaction log.

More information is available on how to create batch files.

See "About using batch files with NetBackup for SQL Server" on page 188.

13 When you have created all the batch files, add these files to the policies that you created previously.

See "Adding the batch files to the policy that protects a specific node in the availability group" on page 139.

Adding the batch files to the policy that protects a specific node in the availability group

To add the batch files to the policy that protects a specific node in the availability group

1 This procedure assumes that you already created a policy and created batch files for a specific node in the AG.

See "Configuring an automatic backup policy for a specific node of a SQL Server availability group" on page 137.

See "Creating a batch file for the policy that protects a specific node in an availability group" on page 138.

2 Open the policy that you created.

3 On the **Backup Selections** tab, add the batch file(s) that you created.

Include batch files for only one type of backup in this policy. (For example, full, full differential, or transaction log.)

4 Click **OK** to save the policy.

5 Repeat the steps in this procedure for each policy that you created.

Configuring SQL Server backups when an availability group crosses NetBackup domains

This procedure describes how to configure NetBackup to back up SQL Server availability group (AG) backups when the AG crosses NetBackup domains. You configure NetBackup to back up a single node in the AG. Then NetBackup replicates that backup to the other NetBackup domains that protect the other AG nodes.

This procedure assumes that:

- One or more nodes in the AG exist in different NetBackup domains.
- The following storage is available in the NetBackup source and target domains that contain the nodes of the AG:
  - For OpenStorage, a disk appliance of the same type in each domain. The disk appliance type must support NetBackup Auto Image Replication (A.I.R.).
  - For NetBackup deduplication, the storage that NetBackup can use for a Media Server Deduplication Pool in each domain.

See the section called “Additional resources” on page 142. for complete details on storage devices, A.I.R., and how to configure these components.

To configure SQL Server backups when an availability group crosses NetBackup domains

1 In each domain where an AG node exists, install and configure a NetBackup master server.

2 In each domain, install and configure any media server(s) that you need.

3 Establish trust from both domains. See the *Security and Encryption Guide* for information on how to establish a trust relationship between master servers for Targeted A.I.R.

4 Choose a node of the AG (or specific replica) from which you want to take backups.

The backups in this domain act as the source backups; this domain acts as the source domain. The other NetBackup domain acts as the target domain and hosts the target storage for replication operations.

5 In each domain, configure a storage server.

- To open the **Storage Server Configuration Wizard**, choose **Media and Device Management > Configure Disk Storage Servers**.

- On the **Welcome** panel, select the type of disk storage that you want to configure.
- Use the wizard to also create a disk pool and storage unit.

6 On the master server in the source domain, open the NetBackup Administration Console.

- Add the target storage servers.
  To perform this configuration, open **Media and Device Management > Credentials > Storage Server**. Edit the storage server and add the target servers to the **Replication** tab.
- Add the target remote master servers to the list of trusted master servers.
  In the host properties for the master server, click the **Servers** node. On the **Trusted Master Servers** tab, add the target remote master servers.

7 On the master server in each of the target domains, create a storage lifecycle policy (SLP), *IMPORT_AG*, to import backup images from the source domain.

The **Import** operation should have the following properties:

Operation	**Import**
Destination storage	Select the storage unit that you want to use
Retention type	**Target Retention** Note that the retention settings in this SLP override the retention settings in the backup policy.

8 On the master server of the source domain, create an SLP, *EXPORT_AG*, to perform a backup and to replicate the backup to the target domains.

The **Backup** operation should have the following properties:

Operation	**Backup**
Destination storage	Select the storage unit that you want to use.
Retention type	**Fixed** Note that the retention settings in this SLP override the retention settings in the backup policy.

The **Replication** operation should have the following properties:

Operation	**Replication**
Send the backups to	**A specific Master Server**
Target master server	Select the target master server.
Target import SLP	Select the *IMPORT_AG* SLP that you created in step 7.

9 For each master server, edit the SLP parameters in the host properties if necessary.

To configure these parameters, expand **Host Properties** and select **Master Servers**. Double-click on the master server you want to edit. Click **SLP Parameters**.

10 On the master server in the source domain, create an MS-SQL-Server backup policy to back up the specific AG node.

- For the **Policy storage**, select the *EXPORT_AG* SLP that you created in step 8.
- Follow the instructions to create a policy for specific node in the AG.
  See “About protecting a specific node in a SQL Server availability group ” on page 136.

Additional resources

NetBackup Administrator’s Guide, Volume I

NetBackup Deduplication Guide

NetBackup OpenStorage Solutions Guide

http://www.netbackup.com/compatibility

Restoring a SQL Server availability group database to the primary and the secondary replicas

In some situations you may need to restore the SQL Server availability group (AG) databases to both the primary and the secondary replica(s). These situations can include when you restore databases:

- Following a disaster recovery
- After logical corruption of the databases
- To a clone of an AG or test environment
- To an earlier point in time

To restore a SQL Server availability group database to the primary and the secondary replicas

1. If you did not already, configure the mappings for distributed application restores.

   Map the WSFC (Windows Server Failover Cluster) name to each AG node. If you have an AG with an FCI, you must configure additional mappings. Configure these mappings in the **Distributed Application Restore Mapping** host property on the master server.

   See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.

2. If you did not already, review the auto-discovered mappings for the hosts in your environment.

   Approve each valid **Auto-Discovered Mapping** that NetBackup discovers in your environment. Perform this configuration in the **Host Management** properties on the master server.

   See "Reviewing the auto-discovered mappings in Host Management" on page 66.

3. Log on to the AG node that hosts the primary replica.
4. Open SQL Server Management Studio and perform the following tasks:
   - Suspend data movement on the database.
   - Remove the database from the AG.
5. Close any connections to the database.
6. Remove the primary database from SQL Server.
7. Open the NetBackup MS SQL Client.
8. Select **File > Set SQL Server connection properties**.

9 From the **Instance** list, select the instance that hosts the AG.

10 Select **File > Restore SQL Server objects**.

You may want to perform this restore for the primary database in parallel with the restore(s) for the secondary database(s).

11 In the **Backup History Options** dialog box, for the **Source Client**, select or type the full qualified domain name (FQDN) of the Windows Server Failover Clustering (WSFC) cluster.

You can find the cluster name in Failover Cluster Manager or the job details for the backup.

NetBackup displays the databases that are included in the availability group. To restore any system or any user databases in the backup, perform a separate browse and restore operation using the node name.

12 Click **OK**.

13 In the **Restore Microsoft SQL Server Objects** dialog box, select the latest full backup image and transaction log backups.

14 Select **Use replace option.**

15 From the **Recovery** list, select **Recovered**.

16 Click **Restore**.

17 When the restore completes, add the database to the AG using the **Skip initial data synchronization** option.

18 Log on to the node that hosts the secondary replica.

19 Close any connections to the database on the secondary replica.

20 Remove the secondary database from SQL Server.

21 Open the NetBackup MS SQL Client.

22 Select **File > Set SQL Server connection properties**.

23 From the **Instance** list, select the instance that hosts the AG.

24 Select **File > Restore SQL Server objects**.

25 In the **Backup History Options** dialog box, for the **Source Client**, select or type the full qualified domain name (FQDN) of the Windows Server Failover Clustering (WSFC) cluster.

You can find the cluster name in Failover Cluster Manager or the job details for the backup.

NetBackup displays the databases that are included in the availability group. To restore any system databases or user databases in the backup, perform a separate browse and restore operation using the node name.

26 Click **OK**.

27 In the **Restore Microsoft SQL Server Objects** dialog box, select the same set of images that you restored to the primary replica.

28 From the **Recovery** list, select **Not recovered**.

29 Select **Use replace option.**

30 If the nodes in the AG use different paths for the database file, you need to create a move template to restore to a secondary replica. From the **Scripting** list, choose **Create a move template**.

See "Performing a SQL Server database move" on page 89.

31 Click **Restore**.

32 When the restore completes, join the database to the AG.

33 Repeat step 18 through step 32 for additional nodes in the AG.

Restoring a SQL Server availability group database to a secondary replica

This procedure describes how to restore a SQL Server availability group (AG) database to a secondary replica. Follow this procedure if a secondary replica is unavailable for an extended time and needs to be synchronized with the primary. Or you can follow these instructions after you add a new secondary replica to the AG.

To restore a SQL Server availability group database to a secondary replica

1. If you did not already, configure the mappings for distributed application restores.

   Map the WSFC (Windows Server Failover Cluster) name to each AG node. If you have an AG with an FCI, you must configure additional mappings.

   Configure these mappings in the **Distributed Application Restore Mapping** host property on the master server.

   See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.

2. If you did not already, review the auto-discovered mappings for the hosts in your environment.

   Approve each valid **Auto-Discovered Mapping** that NetBackup discovers in your environment. Perform this configuration in the **Host Management** properties on the master server.

   See "Reviewing the auto-discovered mappings in Host Management" on page 66.

3. Log on to the node that hosts the secondary replica.
4. Close any connections to the database on the secondary replica.
5. Remove the secondary database from the AG.
6. Open the NetBackup MS SQL Client.
7. Select **File > Set SQL Server connection properties**.
8. From the **Instance** list, select the instance that hosts the AG.
9. Select **File > Restore SQL Server objects**.
10. In the **Backup History Options** dialog box, for the **Source Client**, select or type the full qualified domain name (FQDN) of the Windows Server Failover Clustering (WSFC) cluster.

    You can find the cluster name in Failover Cluster Manager or the job details for the backup.

    NetBackup displays the databases that are included in the availability group. To restore any system databases or user databases in the backup, perform a separate browse and restore operation using the node name.
11. Click **OK**.
12. In the **Restore Microsoft SQL Server Objects** dialog box, select the latest full backup image and transaction log backups.
13. From the **Recovery** list, select **Not recovered**.

14 Select **Use replace option.**

15 If the nodes in the AG use different paths for the database file, you need to create a move template to restore to a secondary replica. From the **Scripting** list, choose **Create a move template**.

See "Performing a SQL Server database move" on page 89.

16 Click **Restore**.

17 When the restore completes, join the database to the AG.

Restoring an availability group database when an availability group crosses NetBackup domains

To restore an availability group (AG) database that was backed up by an AG node in another NetBackup domain, you must first configure NetBackup for Auto Image Replication (A.I.R.). The backup must complete and be replicated to the target node(s). Once the backup is replicated, you can perform a restore on a target node in the same way as you perform any other restore of AG databases.

Note: Replication may not occur immediately to the target AG nodes. The time it takes for replication to occur is dependent on the settings for each master server.

See "Configuring SQL Server backups when an availability group crosses NetBackup domains" on page 140.

See "Restoring a SQL Server availability group database to a secondary replica" on page 145.

See "Restoring a SQL Server availability group database to the primary and the secondary replicas" on page 143.

Configuring backups of clustered SQL Server instances (SQL Server Intelligent Policy)

This procedure describes how to protect SQL Server clustered instances with a SQL Server Intelligent Policy. Perform these steps after you create the virtual SQL Server (VIRTUALSERVER). The following actions must be performed on the master server or on a NetBackup remote client console that acts for the master server.

If you have a SQL Server cluster with multiple NICs, you must follow a different procedure.

See "Configuring backups of a SQL Server cluster when you have multiple NICs (SQL Server Intelligent Policies)" on page 177.

To configure backups of clustered SQL Server instances (SQL Server Intelligent Policy)

1. Open the NetBackup Administration Console.
2. Create a policy (for example, VIRTSQLPOLICY).
3. On the **Attributes** tab, configure the following:
   - Select the **MS-SQL-Server** policy type.
   - Specify a storage unit. If you use a virtual media server, then specify a storage unit that belongs to the virtual media server.
4. On the **Instances and Databases** tab, select **Protect instances**.

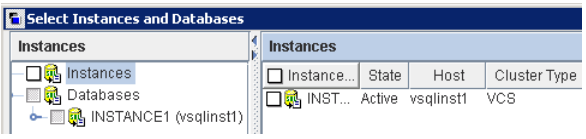
   See "Adding instances to a policy" on page 50.
5. Add the instances or databases that you want to protect.

   See "Adding instances to a policy" on page 50.

   See "Adding databases to a policy" on page 51.

   See "Adding instance groups to a backup policy" on page 57.

   For a clustered instance, the host name is the virtual name of the SQL Server cluster.



6. Add other policy information as follows:
   - Add schedules.
     See "About schedule properties" on page 46.
   - (Optional) Select the specific filegroups or files that you want to back up. By default, NetBackup backs up an entire database.
     See "Adding filegroups or files to the backup selections list" on page 55.
   - (Optional) Make changes to any tuning parameters.
     See "About tuning parameters for SQL Server backups" on page 58.

7 Map the virtual name of the SQL Server cluster to each node in the cluster.

Configure these mappings in the **Distributed Application Restore Mapping** host property on the master server.

See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.

8 Configure the **Mapped Host Names** for the SQL Server hosts in your environment.

Configure this property in Host Management on the master server.

See "Reviewing the auto-discovered mappings in Host Management" on page 66.

Configuring backups of clustered SQL Server instances (legacy SQL Server policies)

This procedure describes how to protect SQL Server clustered instances with a legacy policy that uses batch files and clients. Perform these steps after you create the virtual SQL Server (VIRTUALSERVER). The following actions must be performed on the master server or on a NetBackup remote client console that acts for the master server.

If you have a SQL Server cluster with multiple NICs, you must follow a different procedure.

See "Configuring backups of a SQL Server cluster when you have multiple NICs (legacy SQL Server policies)" on page 178.

To configure backups of clustered SQL Server instances

1 Open the NetBackup Administration Console.

2 Create a policy (for example, VIRTSQLPOLICY).

3 On the **Attributes** tab, configure the following:

- Select the **MS-SQL-Server** policy type.
- Specify a storage unit. If you use a virtual media server, then specify a storage unit that belongs to the virtual media server.

4 On the **Instances and Databases** tab, select **Clients for use with batch files**.

5 On the **Schedules** tab, add an automatic backup schedule.

6 On the **Clients** tab, add the virtual SQL Server name (VIRTUALSERVER).

7 On the **Backup Selections** tab, add one or more script names (batch files).

8 Map the virtual name of the SQL Server cluster to each node in the cluster.

Configure these mappings in the **Distributed Application Restore Mapping** host property on the master server.

See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.

9 Configure the **Mapped Host Names** for the SQL Server hosts in your environment.

Configure this property in Host Management on the master server.

See "Reviewing the auto-discovered mappings in Host Management" on page 66.

Performing a restore of a virtual SQL Server instance

This procedure describes how to perform a restore of a virtual SQL Server instance.

To perform a restore on a virtual SQL Server instance

1 Open the NetBackup MS SQL Client on the active node.

2 Select **File > Restore SQL Server objects**.

3 In the **Backup History Options** dialog box, in the **SQL Host** list, select the virtual server name (VIRTUALSERVER) of the SQL Server.

4 Click **OK**.

5 In the **Restore Microsoft SQL Server Objects** dialog box, select a backup image or staged image list.

6 Click **OK**.

About NetBackup for SQL Server with database mirroring

Note: Database mirroring is not supported for SQL Server Intelligent Policy.

Database mirroring is a software solution that increases the availability of a SQL Server database. It uses two database instances (normally on different hosts), which contain copies of the same SQL Server database. These databases are identical in both name and content. The copies are the principal and the mirror. The

mirror serves as a hot standby to the principal, where transactions take place. The mirror is very closely synchronized with the principal through transaction log porting. It is immediately available in case the principal fails.

The primary consideration when you establish your backup and restore procedures for database mirroring is that these operations are only available on the principal database.

For a complete description of database mirroring refer to the *SQL Server Books Online*.

Configuring NetBackup to support database mirroring

To use database mirroring with NetBackup, both the principal and the mirror should be set up as clients of the same master server.

To configure NetBackup to support database mirroring

1. The hosts that contain both databases should specify the same master server in their server lists.
2. Any policy that is used to back up the principal should also specify the host that contains the mirror database.

   See "Performing simultaneous backups for mirrored partners" on page 152.
3. On the master server, configure permissions for a redirected restored for both mirroring partners.

   See "Configuring permissions for redirected restores" on page 93.
4. (Conditional) If you specify the fully-qualified domain name (FQDN) for the client in the backup policy, you need to create an alias for the short client name. This alias lets you successfully browse for a backup image and restore it in a mirrored environment. NetBackup attempts to find a mirrored partner backup image using the short name of the client host (for example, `client1`). However, the backup image in this case is stored using the FQDN (for example, `client1.domain.com`).

   You can create an alias in one of the following ways:

   - On the NetBackup client, create the following touch file:

     install_path`\dbext\mssql\ClientNameMapping.txt`

     Add an entry `<short name of client host> <FQDN of client host>`. For example:

     `client1 client1.domain.com`
   - On the NetBackup master server, use the `bpclient` command to create the alias:

```
bpclient -client client_name -M master_server -add_alias alias_name
```

For example:

```
bpclient -client client1.domain.com -M master.domain.com -add_alias hpe013-vm02
```

You must use the FQDN for the `-client` argument.

Performing simultaneous backups for mirrored partners

Since backups can occur only on the principal, you must take steps to ensure that you don't miss any scheduled backups due to failover. Establish a procedure to simultaneously initiate backups for both partners, but suppress the operation on the mirror.

When you restore a mirrored database, you must restore it to the node currently in the principal role. See *SQL Server Books Online*.

To simultaneously initiate backups for both partners

1. Create a policy with a backup schedule for the principal.
2. Add the host that contains the mirroring partner to the client list.
3. Create a batch file and add it to the backup selections list.
4. Create a batch file on the mirroring partner that has the same name as the batch file specified in the backup selections policy.

   The batch file on the mirroring partner should be identical to the one used on the principal, with one exception. The value for SQLHOST and SQLINSTANCE are different.

Restoring a mirrored database backup image

Note: Before you restore a mirrored database, you must remove the mirroring attribute.

For mirrored databases, NetBackup can create backup images on either or on both the principal and the mirror server. The **Restore Database** dialog box displays any backups images from both servers. To determine which partner the backup was taken from, look at the property page for the image. To view backup images you can select the **Host name** that contains either of the mirroring partners, provided that NetBackup performed backups for that partner.

For example, assume that mirroring partners are as follows. All of the backups were done on `HostB`, though the principal is currently on `HostA`:

- Principal
  Host name: `HostA`
  SQL Server instance: Solaria
  Database: Accounting
- Mirror
  Host name: `HostB`
  SQL Server instance: Moonbeam
  Database: Accounting

If backup images were created exclusively on `HostA` or on both `HostA` and `HostB`, you can view the images from both partners. Select `HostA` in the **SQL Host** list.

To restore a mirrored backup image

1. Disable mirroring on the principal mirror.

   You can use the appropriate commands in SQL Server Management Studio or use ALTER DATABASE directly.
2. On the principal server, open the NetBackup MS SQL Client.

   When you restore a mirror database, you must run the NetBackup MS SQL Client from the principal server. See *SQL Server Books Online* for information on how to determine which partner is the principal.

   In the previous example, the principal is `HostA`.
3. On the **File** menu, select **Restore SQL Server Objects**.
4. In the **Backup History Options** dialog box, from the **SQL host** list select the mirror server.

   In the previous example, the mirror is `HostB`.
5. Click **OK**.
6. Proceed with the restore as normal.

   NetBackup creates a recovery script for the database that includes images from both partners, as appropriate.

Configuring NetBackup to support database log-shipping

Log shipping is a SQL Server feature that may be employed to enhance the overall availability of your installation. It uses a primary server, which contains the active database, a monitor, and one or more secondary servers. Under log shipping, copies of the transaction log are supplied to the secondary servers on a

per-transaction basis to the secondary servers. This configuration allows each secondary server to be in a standby state in case the primary goes offline.

To use log-shipping with NetBackup, both the primary and the secondary should be set up as clients of the same master server. You must disable log truncation for the transaction log backups.

To configure NetBackup to support database log-shipping

1. The hosts that contain both databases should specify the same master server in their server lists.
2. Any policy that is used to back up the primary should also specify the host that contains the secondary database.

   See "Backing up SQL Server in an environment with log shipping" on page 154.
3. On the master server, configure permissions for redirected restores for both the primary and the secondary server.

   See "Configuring permissions for redirected restores" on page 93.

Backing up SQL Server in an environment with log shipping

Many sites also use the secondary server to off-load certain activities from the primary to minimize its load. However, a backup must *not* be performed on a secondary (or standby) server. Databases must always be backed up on the primary server and restored on the primary server. This requirement is based on the Microsoft SQL Server restriction that is outlined in Microsoft knowledge base article 311115.

If you try to perform a backup on the secondary server, you see a message in the `dbclient` log similar to the following:

```
16:33:26 [1208,2348] <16> CODBCaccess::LogODBCerr: DBMS MSG - ODBC message. ODBC return
code <-1>, SQL State <37000>, Message Text <[Microsoft][ODBC SQL Server Driver][SQL
Server]Database 'Mumbo' is  in warm-standby state (set by executing RESTORE WITH
STANDBY) and cannot be backed up until the entire load sequence is completed.>
```

Chapter 10

Backup and recovery concepts

This chapter includes the following topics:

Overview of SQL Server backup and recovery concepts

A SQL Server instance is created on a Windows host by installing SQL Server. You can install multiple instances on a single host including a default instance and multiple named instances. You can select the database instance that you want to browse and back up. From the NetBackup MS SQL Client, select **File > Set SQL Server connection** properties.

About SQL Server system database types

A SQL Server instance cannot be backed up as a single entity. The largest granularity of a SQL Server backup is the database. SQL Server has system and user databases.

The system databases are as follows:

Master	This database is the "brains" of your installation. It contains a great deal of the metadata that describes your instance. Be sure to retain an up-to-date backup of the master database. Back up the master any time you have made changes to your SQL Server installation, including when you have created or modified other databases. Note that you can only do full database backups on the master. You cannot back up its component files, perform differentials, or backup up its transaction log. Recovery of the master database requires special considerations. See "Preparing for disaster recovery of SQL Server" on page 250.
msdb	The Microsoft SQL Agent uses the msdb for schedules, alerts, and for recording the backup history. All types of backups can be performed on it, providing that it has the full or bulk-load recovery option set.
Model	The model database serves as a template for new databases when the Create Database statement is executed. All types of backups can be performed on it, providing that it has the full or bulk-load recovery option set.
tempdb	The `tempdb` is for the temporary databases that applications use. It cannot be backed up and does not appear in the NetBackup for SQL Server backup browser.

About SQL Server database backups

The following types of backup operations can be performed on databases:

Full	The database, including all of its component files are backed up as a single image. The log file is included in a full database backup. **Note:** The transaction log is not automatically truncated following a full backup. Thus a common practice to preserve disk space is to manually truncate the transaction log following a successful full backup.
Differential	All of the changes since the last full are backed up to a single image.

Transaction log	Transaction log backups are only available for the full and bulk-load recovery options. In this operation, the inactive portion of the transaction log is backed up. The following options are available when you select transaction log backup: ■ Back up and truncate transaction log ■ Back up transaction log, but do not truncate it ■ Back up and restore tail log (legacy SQL Server backup policies) The last option is a backup but it does not create a permanent backup image. A typical use of this type of transaction log backup is: after a page-level restore when the database is recovered, but all of the filegroups have not been brought back on line. By backing up and recovering the tail end of the log, SQL Server is able to bring the database back to a usable state.

About SQL Server filegroup backups

In addition to database backups, you can use separate images to back up the logical filegroups and files that comprise databases.

Filegroups can be backed up in the following ways:

Filegroup backups	A backup can be created from a single filegroup. Scripts for filegroup backups are created when you select individual filegroups in the object browser of the backup database dialog box.
Read-write filegroup backups	This backup contains only the read-write filegroups in a database. If all of the filegroups in a database are set to read-write, then the read-write filegroup backup has the same content as a full database backup. You can create scripts for read-write filegroup backups when you select individual databases and select the "read-write filegroups" type of backup.
Backup of all a filegroup's database files	You effectively back up a filegroup when you back up all of the database files in the filegroup.

Caution: Since the user defines the contents of a partial database backup, NetBackup for SQL Server does not use them for staging recovered backups. So if you rely on NetBackup to stage database recovery for you, the partial backup may not be a good choice.

For SQL Server legacy backup policies, you can also perform partial database backups. In this case, a database backup contains an improvised selection of filegroups that the user can select. You can create a template for partial database

backups when you select individual databases and select the **Create a partial database template** type of backup. The template is created with all of the filegroups commented-out. You can choose the filegroups to include in the partial backup by removing the comments from the filegroups.

About SQL Server differential backups

Differential backups can be created on the full database as well on the filegroup backup units. A differential backup contains the changes to the contents of the object since the last time that it was captured in a full backup.

Note: Note that SQL Server does not let you create a differential on a database file.

What are the components of NetBackup for SQL Server?

Table 10-1 describes the components of NetBackup for SQL Server.

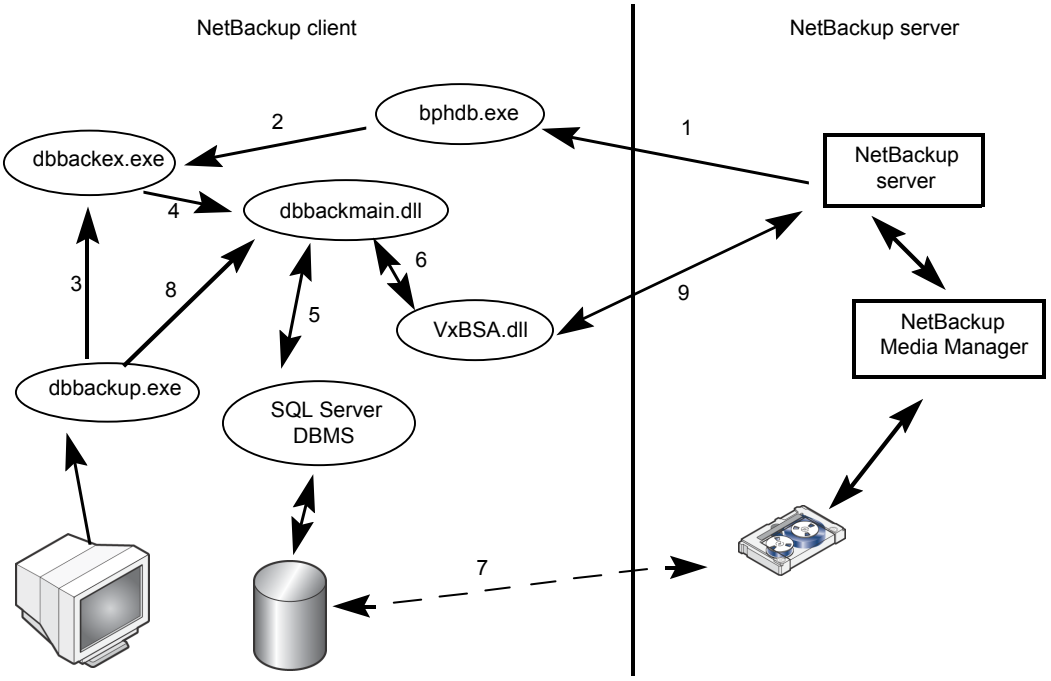
Table 10-1 Components of NetBackup for SQL Server

Component	Filename	Description
graphical user interface (GUI)	dbbackup.exe	You use this interface to: ■ Browse database objects and backup images. ■ Create restore scripts and launch restore operations. ■ (Legacy SQL Server policies) Create backup scripts and launch backup operations.
driver	dbbackex.exe	Launches backup and restore operations.
library	dbbackmain.dll	Facilitates backup and restore activities, access to SQL Server, and other operations that NetBackup for SQL Server performs.

These components also interface with `VxBSA.dll`, which is a common NetBackup client module that connects NetBackup for SQL Server to the NetBackup server.

Figure 10-1 shows the relationships of NetBackup for SQL Server with other software components.

Figure 10-1 NetBackup for SQL Server components



The following interactions occur between NetBackup for SQL Server and other software components:

- Every backup or restore operation is initiated through `dbbackex.exe`, in one of the following ways:
  - Scheduled backups
    The NetBackup scheduler calls `bphdb` (1), which calls `dbbackex` (2).
  - GUI-initiated backups
    `dbbackup.exe` invokes `dbbackex.exe` (3).
  - Command line
    `dbbackex.exe` is invoked directly from a command line or third-party tool.
- `Dbbackex.exe` makes function calls to `dbbackmain.dll` (4) to facilitate a backup or a restore operation. The operation is carried out as `dbbackmain.dll` facilitates one or more data streams between SQL Server and NetBackup server. The data stream (7) is established through VDI (5) and the XBSA interface (6). VDI interacts with SQL Server whereas XBSA interacts with the NetBackup database client.

- (Legacy SQL Server policies) The NetBackup for SQL Server GUI (dbbackup.exe) lets you browse for SQL Server objects, normally, databases, filegroups, and database files. *dbbackup.exe* invokes `dbbackmain.dll` (8) for accessing the SQL Server master database. NetBackup for SQL Server accesses information about SQL Server through ODBC.
- The NetBackup for SQL Server GUI (`dbbackup.exe`) also lets you browse for SQL Server backup images. The NetBackup catalog contains the images you can browse. To access the contents of the catalog the GUI invokes `dbbackmain.dll`, which uses VxBSA function calls to access the NetBackup server database manager.

How does NetBackup resolve SQL Server host and instance names?

Normally SQL Server identifies its installations with a combination that includes the name of the host on which the installation resides plus an instance name. If you omit the instance name then NetBackup assumes that the installation is the default installation on the host. For example, a single host may contain several SQL Server installations, such as, TIGER, TIGER\ACCOUNTING, and TIGER\WAREHOUSE. A clustered instance of SQL Server resides jointly on multiple hosts and is identified with a virtual name.

Host names and SQL Server Intelligent Policies

When you use SQL Server Intelligent Policies, backups are cataloged as follows:

Environment	Catalog name	Example
Instances or databases	Host name that is registered in instance management Usually NetBackup discovers an instance automatically and you register the instance with the NetBackup client name.	sqlhost1
SQL Server cluster	Virtual name of SQL Server	virtsql
Multi-NIC	Private interface name of SQL Server host	sqlhost1-NB
SQL Server cluster in a multi-NIC environment	Private interface name of the virtual SQL Server	virtsql-NB

Host names and legacy SQL Server policies

Environment	Catalog name	Example
Instances or databases	NetBackup client name Usually, host name on which SQL Server resides or the host's NetBIOS name. May also be the IP name (for example, `20.81.74.123`) or the fully qualified domain name (`sqlhost1.mycompany.com`).	sqlhost1
SQL Server cluster	Virtual name of SQL Server	virtsql
Multi-NIC	Private interface name of SQL Server host	sqlhost1-NB
SQL Server cluster in a multi-NIC environment	Private interface name of the virtual SQL Server	virtsql-NB
SQL Server Availability Group (AG)	Fully qualified domain name (FQDN) of the Windows Server Failover Cluster (WSFC)	sql-ag-cluster.mycompany.com

Examples

In most cases when you browse for backup images using the NetBackup Microsoft SQL Client, you only need to specify the **SQL host** name. This setting appears in the **Backup History Options** dialog box.

NetBackup then displays the backup images for all of the instances on that host. However, to ensure that NetBackup displays the backup images you want, consider the following special cases:

- Backups on a network interface that do not have the same name as the host name (such as tiger1 or tiger.apexworks.com)
  In this case, the backup images are stored under the network interface name and not the NetBIOS name. To retrieve these images, see the following instructions:
  See "Performing restores of SQL Server when you have multiple NICs" on page 176.
- Backups from a UNIX (or Linux) server
  This scenario may present a problem because UNIX names are case-sensitive, whereas Windows names are not. In this case, NetBackup tries to retrieve the backup images by specifying the client name with all upper case characters or all lower case characters. If the UNIX client name has mixed uppercase and

lowercase characters, you must provide the client name in the **Source Client** box field.
SQL Host: TIGER
Source Client: Tiger

- The NetBackup client name is a qualified domain name. The SQL Server host name or registered host name (Intelligent Policies) is the NetBIOS name.
  To retrieve backup images specify the **SQL Host** as the NetBIOS name and the **Source Client** as the fully qualified domain name.
  SQL Host: Tiger
  Source Client: tiger.apexworks.com
- The NetBackup client name is an IP address. The SQL Server host name or registered host name (Intelligent Policies) is the NetBIOS name.
  To retrieve backup images specify the **SQL Host** as the NetBIOS name and the **Source Client** as the IP address:
  SQL Host: Tiger
  Source Client: 10.80.136.68
- Backups of a SQL Server cluster
  This scenario does not present an issue because the images are stored under the cluster name. For the **SQL Host name** specify the virtual name of the SQL Server and use the default value for the **Source Client**.

How does NetBackup for SQL Server back up a database?

When a backup is executed, NetBackup for SQL Server does the following: creates a backup script, generates an SQL Server backup statement, logs into SQL Server, and delivers the SQL statement to SQL Server through ODBC. Next, the database agent connects to SQL Server through one or more VDI objects. One virtual device is created per backup stripe. In addition, a VxBSA session is initiated for each stripe. These separate sessions allow NetBackup to start a backup job for each stream that is generated from SQL Server.

When the backup completes, the database agent obtains detailed properties of the object that was backed up, including its relationships to other objects. The agent writes this information to the NetBackup catalog and associates it with the backup image. If there are multiple stripes, then the metadata is associated with the first backup image. The adjunct stripes are associated with one another based upon a common naming convention.

How does NetBackup for SQL Server recover a database?

The NetBackup MS SQL Client displays backup images in a logical hierarchy that mirrors the composition of the database. If you select a transaction log or differential image, then NetBackup examines the metadata that is stored with the images for the selected database. It then determines the most efficient recovery set. Then the agent generates a batch file that includes a sequence of scripted restores. When the scripts are executed, the database is recovered.

The individual restore operations work in a similar manner to backups. A SQL Server restore statement is generated and provided to SQL Server by ODBC. A VDI connection is made. Then a VxBSA session is initiated that starts the data flow between the media manager and SQL Server. NetBackup determines the number of streams (and the corresponding virtual devices and VxBSA sessions) by the number of stripes that were generated during backup.

After all of the recovery operations have completed, the SQL Server agent takes the final step that sets the database into the recovered state. The database goes back online and becomes available for use.

Protecting SQL Server files and filegroups

If your plan to protect SQL Server includes backups of files and filegroups, then the database must use the full or bulk-logged recovery model. In addition, you must maintain the unbroken sequence of transaction log backups. You must create the files and filegroups for your databases and place individual database components into them. NetBackup places a restriction on the layout of your database so it can successfully perform backups and restores of database files and filegroups.

For file or filegroup backups, ensure that a table and its indices reside within the same filegroup.

For example, the layout as indicated by the following Transact SQL statements should not be used:

```
use master
CREATE DATABASE MultiFileDB
ON
PRIMARY ( NAME = FileX,
          FILENAME = 'd:\mssql\data\FileX.mdf'),
FILEGROUP AltGroup
( NAME = AltGroupFil,
  FILENAME = 'd:\mssql\data\AltGroupFil.ndf')
```

```
GO
use MultiFileDB
CREATE TABLE Table1 (col1 char(10),col2 char(10), col3 char(10)) on AltGroup
go
create unique clustered index index4 on Table1 (col2)
go
```

Notice in this example, `Table1` has been placed in filegroup `AltGroup` but its index is placed (by default) in the primary filegroup.

If you place a table into a filegroup that is different than one of its indices, the backup may fail. The following SQL Server error message is displayed:

```
Database file <file name> is subject to logical recovery and
must be among the files to be backed up as part of the file
or filegroup backup.
```

About recovery considerations for SQL Server files and filegroups

Always have backups of a full set of files and filegroups that constitute the entire database. You also need transaction log backups that span the entire period of time over which the backups were taken. When you have both types of backups, it ensures that you can successfully restore a database from file and filegroup backups. To maintain an unbroken sequence of transaction log backups, it is essential to perform a transaction log backup following every file backup or filegroup backup. If you back up several files or filegroups immediately, then you only need to back up the transaction log after the last such backup. If the transaction log is not backed up, SQL Server does not let you restore a file or filegroup.

SQL Server does not keep a record in the transaction log of new files or filegroups that are created. Therefore, after you add either a file or a filegroup to the database, you must immediately back it up. Then perform a backup of all the filegroups in the database so NetBackup selects the correct recovery set when subsequent backups are performed. Similarly, after you create a database file, you should back up all of the files in the filegroup to which it belongs.

Reducing backup size and time by using read-only filegroups

Many applications contain a substantial amount of data that does not change. For example, under time-based partitioning, historical data may be preserved indefinitely.

Only a fraction of the database is subject to change at any given time. Static filegroups can be classified as read-only. NetBackup uses the filegroup read-only designation to optimize the total backup volume speed of recovery.

For backups, the advantage in using read-only filegroups is that you can reduce total media usage. You back up the read-only filegroups one time and retain the backup image indefinitely. This strategy reduces the total time you spend on backup operations because only read-write data is backed up periodically.

For recovery, the advantage is that you can bring your database online more quickly. Read-only filegroups do not need to be restored from backup media unless they are corrupted due to disk error or other hardware failure.

For Intelligent Policy configuration, see the following topics:

See "Backing up read-only filegroups" on page 61.

See "Backing up read-write filegroups" on page 62.

For legacy policy configuration, see the following topics:

See "Backing up read-only filegroups" on page 216.

See "Backing up read-write filegroups" on page 217.

What factors affect the data transfer rate during a SQL Server backup or restore operation?

The following factors can affect the data transfer rate during a SQL Server backup or restore operation: **Maximum transfer size**, **Client buffers per stripe**, **Stripes**, shared memory, and alternate buffer method. For more information, see the following topics and resources:

- See "About tuning parameters for SQL Server backups" on page 58.
- See "About NetBackup for SQL performance factors" on page 70.
- TechNote 33423

About recovery factors for SQL Server

Take into consideration the following factors when you define a recovery plan for your application environment.

Transaction logs	About SQL Server transaction logs
	About backing up the transaction log

Recovery and recovery strategies	About recovery strategies About database recovery About staging recovery
Differential backups	About differential backups
File and filegroup backups	About file and filegroup backups

Much of this information is based on Microsoft's *SQL Server Books Online*. See that resource for a more inclusive discussion.

About SQL Server transaction logs

SQL Server maintains a write-ahead transaction log for each database. This log helps to maintain database updates in cache memory to ensure that data is not written to disk before it has been committed. Database writes occur as a part of the checkpoint procedure.

SQL Server determines the checkpoint frequency based upon the "recovery interval." This interval is a configuration parameter that indicates the maximum time interval that can be tolerated during a system restart. When checkpoint occurs the portion of the transaction log that is no longer needed for system restart becomes inactive and is optionally truncated. The recovery strategy determines whether the transaction log is truncated or not.

See "About recovery strategies" on page 166.

If the checkpoint procedure does not truncate the transaction log, then it can be backed up. Then it can be used for point-in-time recovery, failure from disk crash, or move and copy operations.

About recovery strategies

SQL Server provides the following levels for database recovery. Each level has different implications for both backup performance and for the granularity of recovery.

These levels are as follows:

Simple	With this method you cannot retain the inactive portion of the transaction log beyond the database checkpoint. This method provides for minimal usage of log space. However, the database can only be restored to the last full backup. Transaction log restores, including point in time recovery and marked transaction recovery are not supported. In addition, maximum performance is provided for bulk operations, such as (Create Index, Select Into, and Bulk Copy) because they are not logged.

Full	With this method, the inactive portion of the transaction log is retained until it is truncated, which normally occurs when it is backed up. The transaction log can then be used to stage a recovery either to a point in time or to a marked transaction. The Full Recovery model provides maximum recoverability but it uses the most log space and does not provide maximum performance for bulk operations.
Bulk-Logged	This method is identical to the Full Recovery model except that bulk operations are not logged and thus cannot be recovered.

About backing up the transaction log

By default a transaction log is truncated after it has been backed up. However, it is not truncated following a full database or differential backup. Databases must be set in either full or bulk-logged mode.

The main factors in deciding how frequently to back up a transaction log would be the following:

- Conservation of log space.
- How close to the failure point you must be able to recover in case of a disk crash.

During peak periods in a high transaction environment, it may not be unusual to back up the transaction log every few minutes. (Note that legacy SQL Server policies can only be scheduled to run in hourly increments.)

About differential backups

Unlike the transaction log backup, the differential backup is a backup of the database. The differential includes all of the changes that were made since the last full backup. If you made several differential backups since the last full backup, you only need to restore the last full database, followed by the last differential. You would not need to restore any of the intermediate differentials.

Differential backups include the following types of backups:

- Database differentials
- Individual filegroup differentials
- Read-write filegroup differentials, i.e., any backups that include differentials on all the read-write filegroups in a database.
- Partial differential filegroups, i.e., any backups that include differentials of only the filegroups that the user selects.

Caution: Microsoft recommends that you do not create more than one type of differential backup for the same object.

Caution: NetBackup does not consider differential images when it determines recovery staging strategies if more than one type of differential is found for the same object.

See "About staging recovery" on page 169.

A typical backup procedure may use full database, differential, and transaction log backups in ascending order of frequency. For example the full database backup may be taken bi-weekly and the differential may be taken nightly. Then the transaction log backup may be made more frequently for either mission critical or high volume applications.

About file and filegroup backups

SQL Server also supports the backup of up individual filegroups and files as distinct images. A filegroup is composed of one or more database files. A backup of the constituent files of a filegroup is logically equivalent to a backup of the filegroup itself.

Filegroup and file backups would commonly be used in a tightly architected application in which physical disk locations were mapped to logical objects. For example, tables and indexes.

The following factors may lead you to use file and filegroup backups in this type of environment:

- Some portions of the database should be backed up more frequently than other portions, especially those that may be volatile or mission critical.
- The database may be so large that the time that is required for a full database backup cannot fit in the allocated time window. Thus it may be more viable to do a full backup of one or more files or filegroups on a rotating basis.
- You may want to optimize on backup volume and recovery speed by placing some of your data into read-only filegroups.
  See "Reducing backup size and time by using read-only filegroups" on page 164.

In the event of disk failure, you can choose to recover only the failed unit from a filegroup backup or file backup. You do not have to restore the entire database.

To use filegroup and file backups you must maintain backups of the transaction log.

For example, to perform a full database restore using filegroups and files, you are required to restore the following:

- All of the constituent filegroups and files
- All of the transaction log segments
  These translation logs segments must start from the first component backup until a point in time following the last component backup.

About database recovery

During the restore process, a database goes into "loading mode" until the restore command is executed against the database using the "recovery" option. Before you place the database into recovery mode all of the restore commands are executed using the "Not recovered" option. This way it is possible to continue to stage additional restore statements to bring the database up to the state you want. The database becomes usable again after the last restore statement has been applied the "Recovered" option.

You can choose the recovery option you want when you perform restores.

See “Options for NetBackup for SQL Server restores” on page 81.

About staging recovery

NetBackup for SQL Server keeps track of the backups you have performed and when you performed them. You can display the backup history by opening the Restore Microsoft SQL Server Objects dialog box.

See “Options for NetBackup for SQL Server restores” on page 81.

This dialog box depicts all of the SQL Server backup images within the parameters that you specify. The images appear in a tree-form that is based on the backup types that you performed.

When you select a transaction log for restore, NetBackup for SQL Server automatically searches for a set of images. The images are used to stage a full database recovery. The recovery set consists of the selected transaction log image plus additional images which can reconstitute the database to a recovered state.

A recovery set can include a full database image, filegroup and file images, and differentials images of the database or filegroups. It can include the filegroups that contained partial and read-write images or filegroup differentials that are contained in these images. If the recovery set contains filegroup, file, partial, or read-write images, then it also contains one or more transaction log images.

A read-write filegroup is also a full recovery set but it only contains backups (including differential backups) of those filegroups that are writable. Read-only

filegroup images are not required because they are assumed not to have changed. A read-write filegroup recovery set also contains one or more transaction log images.

If a full recovery set is found, then all of the composite images are checked. In addition, **Stage full recovery** is enabled. To view the full recovery set, right-click the transaction log, select **Properties**, and click the Recovery Set tab.

Chapter 11

Using NetBackup for SQL Server with multiple NICs

This chapter includes the following topics:

About configuration of SQL Server backups with multiple NICs

Many administrators want to reserve a separate network interface for their SQL Server host machines that are used for routing backup traffic. This type of

environment requires additional configuration for backup policies and the NetBackup client that backs up SQL Server. Special configuration is also required to perform restores.

Note: In NetBackup 8.1, if you have a SQL Server cluster in a private network additional configuration is required. You must configure the mappings for distributed application restores. You also must review the auto-discovered mappings for the hosts in your environment.

See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.

See "Reviewing the auto-discovered mappings in Host Management" on page 66.

The following distinct network resources exist in a multi-NIC environment:

- The public name of each SQL Server host (for example, `sqlhost1` and `sqlhost2`)
- The private interface name that is used to back up each of the SQL Server hosts (for example, `sqlhost1-NB` and `sqlhost2-NB`)

The following additional resources exist for a SQL Server cluster in a multi-NIC environment:

- The public virtual name of the SQL Server (for example, `virtsql`)
- The private virtual name of the SQL Server (for example, `virtsql-NB`)

The following requirements exist to use NetBackup for SQL Server in a multi-NIC environment:

- Install the NetBackup client on the SQL Server using the private name of the SQL Server host as the NetBackup client name.
  Alternatively, you can configure the NetBackup client name after installation.
  See "Configuring the NetBackup client with the private interface name" on page 173.
- Configure a backup policy that includes the private interface name of the host or client.
  See "Configuring backups of SQL Server when you have multiple NICs (SQL Server Intelligent Policies)" on page 174.
  See "Configuring backups for SQL Server when you have multiple NICs (legacy SQL Server policies)" on page 175.
  See "Configuring backups of a SQL Server cluster when you have multiple NICs (SQL Server Intelligent Policies)" on page 177.
  See "Configuring backups of a SQL Server cluster when you have multiple NICs (legacy SQL Server policies)" on page 178.

Note that if you want to protect a SQL Server cluster with a legacy SQL Server policy, you must edit the backup batch file. The `BROWSECLIENT` parameter must indicate the private name of SQL Server host or virtual SQL Server.

- For restores in a multi-NIC environment, refer to the following topic:
  See "Performing restores of SQL Server when you have multiple NICs" on page 176.
  If you want to perform a restore from a SQL Server cluster, you must edit the restore batch file. In the batch file, you must change the `BROWSECLIENT` parameter to indicate the private name of virtual SQL Server.
  See "Performing restores of a SQL Server cluster when you have multiple NICs" on page 180.

Configuring the NetBackup client with the private interface name

To perform backups over a private network interface, NetBackup must use the private name of the client. If you installed the NetBackup client using the public interface name, follow this procedure to configure the NetBackup client name as the private interface name.

For cluster environments, additional configuration is required. In that case, NetBackup must use the private virtual name of the SQL Server cluster.

See "Configuring backups of clustered SQL Server instances (legacy SQL Server policies)" on page 149.

To configure the NetBackup client with the private interface name

1. Open the Backup, Archive, and Restore interface.
2. Select **File > NetBackup Client Properties**.
3. Click the **General** tab.
4. In the **Client name** box, specify the private name of the client.

   For example, the private name for the computer `sqlhost1` is `sqlhost1-NB`.

Configuring backups of SQL Server when you have multiple NICs (SQL Server Intelligent Policies)

This topic describes how to create a SQL Server Intelligent Policy to protect a SQL Server when you have multiple NICs. The following configuration changes must be made to allow for backups and restores over a private interface:

- Install the NetBackup client on the SQL Server using the private name of the SQL Server host as the NetBackup client name.
  Alternatively, you can configure the NetBackup client name after installation.
  See "Configuring the NetBackup client with the private interface name" on page 173.
- The backup policy must include the private interface name of the SQL Server host.
  During instance discovery NetBackup automatically adds an instance with the NetBackup client name. If you installed the NetBackup client using the private interface name, NetBackup uses the private name when it performs backups.

To configure a backup policy for a SQL Server in a cluster with a multi-NIC (SQL Server Intelligent Policies)

1. If you installed the NetBackup client on the SQL Server host using the public interface name, follow the procedure to configure the NetBackup client name as the private interface name.

   See "Configuring the NetBackup client with the private interface name" on page 173.
2. Open the NetBackup Administration Console.
3. Expand **NetBackup Management > Applications > Microsoft SQL Server**.
4. Click **All Instances**.
5. Find and register the instance that has the private interface name of the SQL Server host (sqlhost1-NB).
6. Create a new policy or open an existing policy.
7. On the **Instances and Databases** tab, select **Protect instances**.
8. Click **New**.

9 To add the instances or databases that you want to protect, select or expand the instance that has the private interface name of the SQL Server (`sqlhost1-NB`).

See "Adding instances to a policy" on page 50.

See "Adding databases to a policy" on page 51.

10 Add other policy information as follows:

- Add schedules.
  See "About schedule properties" on page 46.
- Add database objects to the backup selections list.
  See "Adding filegroups or files to the backup selections list" on page 55.
- (Optional) Make changes to any tuning parameters.
  See "About tuning parameters for SQL Server backups" on page 58.

Configuring backups for SQL Server when you have multiple NICs (legacy SQL Server policies)

This topic describes how to configure a legacy backup policy using batch files to protect SQL Server with a multi-NIC. The following configuration changes must be made to allow for backups and restores over a private interface:

- Install the NetBackup client on the SQL Server using the private name of the SQL Server host as the NetBackup client name.
  Alternatively, you can configure the NetBackup client name after installation.
  See "Configuring the NetBackup client with the private interface name" on page 173.
- The backup policy must include the private interface name of the SQL Server host.

To configure backups for SQL Server when you have multiple NICs (legacy backup policies)

1 If you installed the NetBackup client on the SQL Server host using the public interface name, follow the procedure to configure the NetBackup client name as the private interface name.

See "Configuring the NetBackup client with the private interface name" on page 173.

2 Open the NetBackup Administration Console.

3 Create a new policy or open an existing policy.

4 On the **Clients** tab, add a new client.

For the Client name, provide the private interface name. For example, the public name is `sqlhost1`. The private interface that is used to back up `sqlhost1` is `sqlhost1-NB`.

5 Add other policy information as follows:

- Add schedules.
  See "About schedule properties " on page 202.
- Create and add batch files to the backup selections list.
  See "About using batch files with NetBackup for SQL Server" on page 188.
  See "Adding batch files to the backup selections list " on page 208.

Performing restores of SQL Server when you have multiple NICs

To perform restores of a SQL Server in a multi-NIC environment, you need to do the following:

- Connect to SQL Server host using the public name of the host.
- To browse for backup images, specify public name of the SQL Server for the **SQL Host** name. Specify the private name of the SQL Server for the **Source Client**.

If you use SQL Server policies in a cluster environment, you must follow a different procedure:

See "Performing restores of a SQL Server cluster when you have multiple NICs" on page 180.

To perform SQL Server restores when you have multiple NICs

1 Open the NetBackup MS SQL Client.

2 Select **File** > **Set SQL Server connection properties**.

3 In the **Host** box, specify the public name of the SQL Server host.

4 Click **OK**.

5 Select **File > Restore SQL Server objects**.

6 In the **SQL Host** box, specify the public name of the SQL Server host (`sqlhost1`).

7 In the **Source Client** box, specify the private interface name of the SQL Server host (`sqlhost1-NB`).

8 Click **OK**.

A dialog box opens that shows the SQL Server backups that the **SQL Host** made on the private network interface.

9 Continue with the restore as normal.

See "Restoring a SQL Server database backup" on page 84.

Configuring backups of a SQL Server cluster when you have multiple NICs (SQL Server Intelligent Policies)

This topic describes how to create a SQL Server Intelligent Policy to protect a SQL Server when you have multiple NICs. During instance discovery NetBackup automatically adds an instance with the NetBackup client name. For a virtual SQL Server in a multi-NIC environment, you must add and register the instance with the private interface name of the virtual SQL Server. This name is the instance name that you add to the backup policy.

To configure a backup policy for a SQL Server cluster with a multi-NIC (SQL Server Intelligent Policies)

1 Open the NetBackup Administration Console.

2 Expand **NetBackup Management > Applications > Microsoft SQL Server**.

3 Click **All Instances**.

4 Manually add a new instance and register it. For the **Host**, provide the private interface name of the virtual SQL Server (`virtsql-NB`).

5 Create a new policy or open an existing policy.

6 On the **Instances and Databases** tab, select **Protect instances**.

7 Click **New**.

8 To add the instances or databases that you want to protect, select or expand the instance that has the private interface name of the virtual SQL Server (`virtsql-NB`).

See "Adding instances to a policy" on page 50.

See "Adding databases to a policy" on page 51.

9 Add other policy information as follows:

- Add schedules.
  See "About schedule properties" on page 46.

- Add database objects to the backup selections list.
  See “Adding filegroups or files to the backup selections list” on page 55.
- (Optional) Make changes to any tuning parameters.
  See “About tuning parameters for SQL Server backups” on page 58.

Configuring backups of a SQL Server cluster when you have multiple NICs (legacy SQL Server policies)

This topic describes how to create a legacy SQL Server backup policy to protect a SQL Server cluster with a multi-NIC. When you create the backup policy, it must include a client that has the private interface name of the virtual SQL Server. The public name of the host should not be used.

To configure backups of a SQL Server when you have multiple NICs (legacy backup policies)

1. Open the NetBackup Administration Console.
2. Create a new policy or open an existing policy.
3. On the **Clients** tab, add a new client.

   For the client name, use the private interface name of the virtual SQL Server. For example, `virtsql-NB`.
4. Add other policy information as follows:
   - Add schedules.
     See “About schedule properties ” on page 202.
   - Create a batch file that includes the private interface name of the virtual SQL Server. Then add this batch file to the backup selections list.
     See “Creating a batch file for backups of a SQL Server cluster when you have multiple NICs (legacy SQL Server policies)” on page 178.
     See “Adding batch files to the backup selections list ” on page 208.

Creating a batch file for backups of a SQL Server cluster when you have multiple NICs (legacy SQL Server policies)

This topic describes how to create a batch file for a legacy backup policy to protect a SQL Server cluster with a multi-NIC connection. To create the batch file you need

to connect to the SQL Server host using the public name of the virtual SQL Server. The batch file must include the private name of the virtual SQL Server.

To create a batch file for SQL Server cluster backups with a multi-NIC connection

1. On any node in the SQL Server cluster, open the NetBackup for SQL Server interface.
2. Select **File > Set SQL Server connection properties**.
3. In the **Host** box, specify the public name of the virtual SQL Server host (`virtsql`).
4. Click **Apply** and **Close**.
5. Select **File > Backup SQL Server objects**.
6. Select the databases to back up.
7. Select the backup options.

   Note: Do not attempt to perform an immediate backup from the backup dialog box. The generated batch files must be modified before they can be run successfully.

8. From the **Backup script** options, click **Save**.
9. Click **Backup**.

   A batch file similar to the following is created:

```
OPERATION BACKUP
DATABASE "ACCOUNTING"
SQLHOST "VIRTSQL"
NBSERVER "THOR"
BROWSECLIENT "VIRTSQL"
MAXTRANSFERSIZE 0
BLOCKSIZE 7
ENDOPER TRUE
```

10 Change the line value that is associated with the `BROWSECLIENT` from the public name of the virtual SQL Server to the private name.

```
OPERATION BACK
UPDATABASE "ACCOUNTING"
SQLHOST "VIRTSQL"
NBSERVER "THOR"
BROWSECLIENT "VIRTSQL-NB"
MAXTRANSFERSIZE 0
BLOCKSIZE 7
ENDOPER TRUE
```

11 Place the modified batch file on all nodes in the cluster or in a shared location. This way it is available for scheduled backups.

Backups are done regardless of which node is active when a backup is initiated.

Performing restores of a SQL Server cluster when you have multiple NICs

To perform restores of a SQL Server cluster in a multi-NIC environment, you need to do the following:

- Connect to virtual SQL Server host using the public name of the host.
- To browse for backup images, specify public name of the virtual SQL Server for the **SQL Host** name. Specify the private name of the virtual SQL Server for the **Source Client**.
- Create a batch file for the restore and manually edit it to include the private name of the virtual SQL Server.

If you do not have a cluster environment, you must follow a different procedure:

See "Performing restores of SQL Server when you have multiple NICs" on page 176.

To perform restores of a cluster when you have multiple NICs

1 On a specific node in the cluster, open the NetBackup for SQL Server interface.

2 Select **File** > **Set SQL Server connection properties**.

3 In the **Host** box, specify the public name of the virtual SQL Server host (`virtsql`).

4 Click **Apply** and **Close**.

5 Select **File** > **Restore SQL Server objects**.

6 In the **Backup History Options** dialog box, specify the following.

SQL Host	Public name of the virtual SQL Server (`virtsql`).
Source Client	Private name of the virtual SQL Server (`virtsql-NB`).

7 Click **OK**.

8 Select the databases to restore.

See "Options for NetBackup for SQL Server restores" on page 81.

Note: Do not try to perform an immediate restore from the restore dialog box. The generated batch files must be modified before they can be run successfully.

9 Select the restore options.

10 From the **Restore script options**, select **Save**.

11 Click **Restore**.

The NetBackup MS SQL Client generates a batch file that is similar to the following.

```
OPERATION RESTORE
OBJECTTYPE DATABASE
DATABASE "ACCOUNTING"
NBIMAGE "SQLHOST1.MSSQL7.VIRTSQL.db.ACCOUNTING.~.7.001of001.20040306111309..C"
SQLHOST "VIRTSQL"
NBSERVER "THOR"
BROWSECLIENT "VIRTSQL"
MAXTRANSFERSIZE 0
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE RECOVERED
ENDOPER TRUE
```

12. Change the line value that is associated with `BROWSECLIENT` from the public name of the virtual SQL Server to the private name.

```
OPERATION RESTORE
OBJECTTYPE DATABASE
DATABASE "ACCOUNTING"
NBIMAGE "SQLHOST1.MSSQL7.VIRTSQL.db.ACCOUNTING.~.7.001of001.20040306111309..C"
SQLHOST "VIRTSQL"
NBSERVER "THOR"
BROWSECLIENT "VIRTSQL-NB"
MAXTRANSFERSIZE 0
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE RECOVERED
ENDOPER TRUE
```

13. Select **File** > **Manage script files**.
14. Select the modified batch file and click **Start**.

Chapter 12

Configuring backups with legacy SQL Server policies using clients and batch files

This chapter includes the following topics:

- Backing up SQL Server transaction logs
- Backing up SQL Server database filegroups
- Backing up read-only filegroups
- Backing up read-write filegroups
- Backing up SQL Server database files
- Performing partial database backups
- Performing a backup of a remote SQL Server installation
- About file checkpointing with NetBackup for SQL Server
- About automatic retry of unsuccessful SQL Server backups

About legacy SQL Server policies

A legacy NetBackup for SQL policy includes a list of SQL Server database clients and a batch file that contains SQL Server backup commands. When a backup is scheduled, NetBackup runs the commands in the batch file for each client in the policy. You create the batch file through the NetBackup MS SQL Client interface, which saves the options you select to a batch file. Or you can create this batch file manually.

The legacy SQL Server policy includes the following criteria:

- Storage unit and media to use
- Policy attributes
- Backup schedules: Automatic schedule (called Full Backup) and application schedule
- Clients to be backed up
- Backup batch files to be run on the clients

Use the NetBackup Administration Console to configure a legacy backup policy. The Policy utility contains all the settings and parameters you need to create or change this kind of policy.

The Policy Configuration Wizard is not available for legacy SQL Server policies.

About configuring backups with legacy SQL Server policies

Table 12-1 Steps to configure SQL Server backups that use legacy SQL Server policies

Step	Action	Description
Step 1	Verify that you have a supported SQL Server configuration.	See the Application/Database Agent Compatibility List.
Step 2	Ensure that requirements are met for the NetBackup server and the SQL Server software.	See "NetBackup server and client requirements" on page 21.
Step 3	Configure the logon account for the NetBackup services.	The logon account for the NetBackup Client Service and the NetBackup Legacy Network Service must meet certain requirements. See "Configuring the NetBackup services for SQL Server backups and restores" on page 27.
Step 4	Review the information on how NetBackup for SQL Server uses SQL Server security.	See "About SQL Server security with NetBackup legacy backup policies" on page 187.
Step 5	Configure the batch files for the policy.	See "About using batch files with NetBackup for SQL Server" on page 188.
Step 6	Configure a legacy SQL Server policy.	See "Adding a new SQL Server legacy policy" on page 201.
Step 7	If you have a SQL Server AG or cluster, you must configure the mappings for distributed application restores.	Configure these mappings in the **Distributed Application Restore Mapping** host property on the master server. See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.
Step 8	If you have a SQL Server AG or cluster, you must review the auto-discovered mappings for the hosts in your environment.	Approve each valid **Auto-Discovered Mapping** that NetBackup discovers in your environment. Perform this configuration in the **Host Management** properties on the master server. See "Reviewing the auto-discovered mappings in Host Management" on page 66.

Configuring the NetBackup services for SQL Server backups and restores

NetBackup uses the NetBackup Client Service and the NetBackup Legacy Network Service to access the SQL Server when it performs backups and restores. With the proper configuration, these services can log on with the Local System account or another account that has the necessary privileges.

The logon account for the services requires the following:

- Both services must use the same logon account.
- The SQL Server “sysadmin” role.
- If you want to use Local System for the logon account, the requirements depend on the SQL Server version:
  - For SQL Server 2008, the sysadmin role is automatically applied to the NT AUTHORITY\SYSTEM and BUILTIN\Administrators groups.
  - For SQL Server 2012 and later, you must first apply the sysadmin role manually to the NT AUTHORITY\SYSTEM or the BUILTIN\Administrators group.
- For a SQL Server cluster or SQL Server availability group (AG), configure the NetBackup services on each node in the cluster or AG.
- For VMware backups, different configuration is required for logon account for the services.
  See “Configuring the NetBackup services for VMware backups that protect SQL Server” on page 106.

To configure the NetBackup services for SQL Server backups and restores

1. Log on to the Windows host with the account that has the sysadmin role.
2. If the SQL Server instance uses standard or mixed security, perform the following steps:
   - Open the NetBackup MS SQL Client.
   - Select **File > Set SQL Server connection properties**.
   - Provide the SQL Server **Userid** and **Password**.
   - Click **Apply**.
   - Click **Close**.
3. Open the Windows Services application.
4. Double-click the **NetBackup Client Service** entry.

5 Click the **Log On** tab.

6 Confirm that **Local System account** is selected.

7 Click **OK**.

8 If you selected a different logon account, stop and restart the service.

9 Double-click the **NetBackup Legacy Network Service** entry.

10 Click the **Log On** tab.

11 Confirm that **Local System account** is selected.

12 Click **OK**.

13 If you selected a different logon account, stop and restart the service.

About SQL Server security with NetBackup legacy backup policies

NetBackup for SQL Server uses SQL Server backup and restore commands and queries the SQL master database. These operations are validated according to the security method you choose when you install SQL Server, either integrated security or standard security. Integrated security refers to the use of Windows authentication in lieu of standard SQL Server-based logons.

Note: Microsoft recommends using integrated security. Unlike SQL Server-based logons, Windows logons can be traced with standard Windows security tools. NetBackup for SQL Server supports both integrated security and standard security for any level of SQL Server.

If you use integrated security, the Windows account you log into is used for authentication. SQL Server ignores any user ID and password that you enter in the NetBackup MS SQL Client or in a batch file.

If you use standard security, then you must supply a SQL Server-based user ID and password. Once you provide these credentials, NetBackup stores this information in the registry (the password is encrypted) under the following registry key:

```
HKEY_CURRENT_USER\SOFTWARE\VERITAS\NETBACKUP\NetBackup for
Microsoft SQL Server\
```

About using batch files with NetBackup for SQL Server

NetBackup for SQL Server uses batch files to initiate backup and restore operations. A batch file uses the `.bch` extension and is typically executed from the *`install_path`*`\DbExt\MsSql\` directory.

You must create a batch file if you start operations in any of the following ways:

- Manage Scripts dialog box
- `dbbackex` command line
- Automatically scheduled backups that use batch files and clients

Rules for using batch files

Review the following information before you create and use batch files:

- Ensure that the batch file resides on the client.
  See "Registering authorized locations used by a NetBackup database script-based policy" on page 268.
- Batch files are in Unicode text.
- A batch file consists of a series of operations that run in sequence. For legacy SQL Server backup policies, you create batch files for backup operations and restore operations. For SQL Server Intelligent Policy, you create the batch files for restore operations in the same way.
- Each operation consists of a series of *<keyword value>* pairs, which completely define the total operation.
- The keyword is not case-sensitive but the value is. Generally, you can code both the keyword and value in uppercase. The exception is the `NBIMAGE` keyword option. The value must be specified exactly as it appears in the NetBackup server.
- Operations are not nested.
- With the exception of the `BATCHSIZE`, `GROUPSIZE`, `RESTARTTYPE`, `NUMRESTARTS`, and `RESTARTWAITSECONDS` parameters, *<keyword value>* pairs are not global. If you use `BATCHSIZE`, `GROUPSIZE`, `RESTARTTYPE`, `NUMRESTARTS`, or `RESTARTWAITSECONDS` then it must appear only once in your batch file and it must appear in the first operation.
- If `SQLINSTANCE $ALL` is used, then it must appear in the first operation of the batch file. Each operation in the batch file is performed for all SQL Server

instances on the client where the batch file is executed. Also, it is not necessary to specify an SQLHOST or SQLINSTANCE on any subsequent operations.

- Within an operation, the *<keyword value>* pairs may appear in any order except that you must terminate each operation with ENDOPER TRUE.
- You can include comment lines in your batch file by placing a hash mark ('#') in the first column.
- STOPAT, RESTORETOMARK, RESTORETOMARKAFTERTIME, RESTOREBEFOREMARK, and RESTOREBEFOREMARKAFTERTIME are mutually exclusive restore parameters. If either RESTORETOMARKAFTERTIME or RESTOREBEFOREMARKAFTERTIME are used, then the batch file must also specify a datetime string with the keyword STOPAFTER.
- If you remove the MAXTRANSFERSIZE keyword from the batch file, the default is 0 or a maximum transfer size of 64 KB. If you remove the BLOCKSIZE keyword from the batch file, the default is 0 or a block size of .5 KB. A default value of 0 is also applied if you manually create a batch file without these keywords.

Keywords and values used in batch files

Table 12-2 describes the keywords and values that can be used in batch files.

Table 12-2 Keywords and values used in batch files

Keyword	Values	Required?	Default	Description
ALTCLIENT (Same as BROWSECLIENT)	string	no	none	Restores the images from a host other than the local host.
BACKUPMODEL	BACKUPMODEL_ CONVENTIONAL, BACKUPMODEL_ SNAPSHOT	no	BACKUPMODEL_ CONVENTIONAL	Valid only for restore. Indicates whether the backup was originated from a snapshot method
BATCHSIZE	integer	no	1	Number of backup operations to start simultaneously, per database instance. Applies to all of the operations in the batch file. Must appear before the end of the first operation. Range is 1-32.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
BLOCKSIZE	integer	no	0	Applicable for backup operations only. Block size is calculated as 512 bytes * $2^{BLOCKSIZE}$. Range is 0-7.
BROWSECLIENT (Same as ALTCLIENT)	string	no	none	Restores the images from a host other than the local host.
BUFFERS				See NUMBUFS.
CONSISTENCYCHECK	FULLINCLUDINGINDICES, FULLEXCLUDINGINDICES, PHYSICALCHECKONLY, CHECKCATALOG	no	none	Performs the specified consistency check after the restore has been completed.
CONVERTBACKUP	TRUE, FALSE	no	FALSE	If no previous full backup exists for the database or filegroup, then NetBackup converts the differential or log backup to a full backup. This option also detects if a full recovery database was switched to the simple recovery model and back to the full recovery model. In this scenario, the log chain is broken and SQL Server requires a differential backup before a subsequent log backup can be created. If NetBackup detects this situation, the backup is converted to a differential database backup. See "Converting differential backups to full backups" on page 204.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
COPYONLY	TRUE, FALSE	no	See description	If TRUE, SQL Server creates an out-of-band backup so that it does not interfere with the normal backup sequence. The default value is FALSE except for full database Instant Recovery backups. See "Using copy-only snapshot backups to affect how differentials are based" on page 121.
DATABASE	string	yes	none	Name of database. For backup operations, specify value $ALL to designate all databases (except for tempdb.)
DBMS	MSSQL	no	MSSQL	You can specify MSSQL only.
DSN	string	no	saved from GUI user session	ODBC data source name. Deprecated.
DUMPOPTION	INCREMENTAL	no	none	Specifies INCREMENTAL restoring from an incremental backup.
ENABLESERVICEBROKER	TRUE	no	none	Enables SQL Server Service Broker after a restore operation. To take effect, RECOVERED STATE must be set to RECOVERED. Include this keyword in each individual RESTORE operation.
ENDOPER	TRUE	yes	none	Terminates each operation that is specified in the batch file.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
EXCLUDE	string	no	none	Name of a database to exclude when DATABASE $ALL is specified in a batch operation EXCLUDE can be used in a batch file only if DATABASE $ALL is used.
GROUPSIZE	integer between 1 and 32	no	none	The number of databases that are snapped as a single SQL Server backup image. (Legacy policies) For availability group backups, all databases in the grouped backup must be part of the availability group. NetBackup does not support any grouped snapshot backups that include both AG and non-AG databases. (Intelligent Policies) NetBackup does not support grouped snapshot backups.
INHIBITALTBUFFERMETHOD	TRUE, FALSE	no	FALSE	Tells NetBackup whether to consider the candidacy of alternate buffer method.
MAXRESTARTSETS	integer	no	none	Use MAXRESTARTSETS to enable file checkpointing. The valid range is 2 to 32. This parameter specifies the number of separate streams into which the backup request is sub-divided.
MAXTRANSFERSIZE	integer	no	0	Maximum transfer size is calculated as 64 kilobytes bytes * $2^{MAXTRANSFERSIZE}$. Range is 0-6.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
MOVE	filegroup	no	none	Specifies a filegroup name. Used for the MOVE restore type. For any backups that were made with a SQL Server legacy policy, the PARTIAL restore type also applies.
NBIMAGE	string	yes*	none	Specifies a NetBackup image for the restore operations. See note for NBSERVER. * Required for restore operations.
NBSCHED	string	no	none	If the NetBackup policy has several Application Backup Policy schedules, use NBSCHED to select amongst them.
NBSERVER	string	no	none	Specifies which master server to use for the backup or restore operation. **Note:** If NBSERVER is not specified in a batch file operation, the master server defaults to the name that is specified at HKEY_CURRENT_USER\Software\VERITAS\NetBackup\NetBackup for Microsoft SQL Server\DEFAULT_SQL_NB_MASTER_SERVER.
NUMBUFS	integer	no	1	Number of buffers per stripe. Range is 1-32.
NUMRETRIES				See NUMRESTARTS.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
NUMRESTARTS	1-9	no	1	The number of times to retry a backup if RESTARTTYPE AUTO is specified. Use this keyword only once in the batch file and in the first operation of the batch file.
OBJECTNAME	string	yes*	none	Specifies a file or a filegroup name for file or for filegroup backups and restores, * If OBJECTTYPE= FILE or FILEGROUP.
OBJECTTYPE	DATABASE, TRXLOG, FILEGROUP, FILE	no	DATABASE	Specifies the object you want to back up or restore, a database, transaction log, filegroup, or file.
OPERATION	BACKUP, RESTORE	no	BACKUP	Type of operation, either backup or restore.
PAGE	Page ID	no	none	Ignored for a restore if the backup was performed with SQL Server Intelligent Policy. Specifies a page ID for a page restore operation.
PARTIAL	TRUE, FALSE	no	FALSE	Ignored for a restore if the backup was performed with SQL Server Intelligent Policy. Specifies NetBackup perform a partial backup or restore.
PASSWORD	string	no	null	Password for logging into SQL Server. This keyword is ignored if you use integrated security.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
RECOVERED STATE	RECOVERED, STANDBY, NOTRECOVERED, TRUE, FALSE	no	RECOVERED	RECOVERED means that the database should be restored to the recovered state. NOTRECOVERED means that it should remain in the loading state following the restore. STANDBY means that the database should be restored to standby state. If STANDBY is used, then the STANDBYPATH keyword is also required. TRUE and FALSE, when used as values for RECOVEREDSTATE, are synonyms for RECOVERED and NOTRECOVERED.
RESTARTTYPE	AUTO, MANUAL	no	none	Available only for backups. Use AUTO to automatically retry backup of failed objects. Use MANUAL to create a batch file for backing up any of the objects that were not successfully backed up. Use this keyword only once in the batch file and in the first operation of the batch file.
RESTARTWAITSECONDS	integer number	no	60	The time to make a second attempt following a backup failure. Use this keyword only once in the batch file and in the first operation of the batch file.
RESTOREBEFOREMARK	string	no	none	Specify transaction log mark.
RESTOREBEFOREMARK AFTERTIME	string	no	none	Specify transaction log mark.
RESTOREOPTION	REPLACE	no	none	Tells NetBackup to use the SQL Server replace option on a restore.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
RESTOREPAGES	TRUE, FALSE	no	FALSE	Ignored for a restore if the backup was performed with SQL Server Intelligent Policy. Specifies that NetBackup perform a page restore operation.
RESTORETOMARK	string	no	none	Specify transaction log mark.
RESTORETOMARK AFTERTIME	string	no	none	Specify transaction log mark.
RESTORETYPE	FULL, PARTIAL, MOVE	no	FULL	Full = Full database restore, Move = Database move RESTORETYPE is applicable only to RESTORE database operations. If MOVE is used, then the batch file should contain a series of one or more <MOVE><filegroup> and <TO><file path> sequences. (SQL Server legacy policies only) Partial = Partial database restore. If PARTIAL is used, the sequence for PARTIAL must specify all of the filegroups in the database whose backup image is referenced by the NBIMAGE keyword.
RETRYTYPE				See RESTARTTYPE.
RETRYWAITSECONDS				See RESTARTWAITSECONDS.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
ROLLBACKVOLUME	TRUE, FALSE	no	FALSE	Tells NetBackup to do the recovery of an Instant Recovery backup using the volume rollback method.
SQLCOMPRESSION	TRUE, FALSE	no	FALSE	Uses SQL Server compression on the backup image. If you enable SQL Server compression, do not enable NetBackup compression.
SQLHOST	string	no		Name of SQL Server host. If SQLHOST is not specified in a batch file operation, then the SQL Server host is obtained from HKEY_CURRENT_USER\Software\VERITAS\NetBackup\NetBackup for Microsoft SQL Server\DEFAULT_SQL_HOST. If the SQLINSTANCE keyword is not included, then the default SQL Server instance is assumed for the SQL Host.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
SQLINSTANCE	string	no		Name of the SQL Server instance. Or for backup operations specify $ALL to designate all SQL Server instances including the default instance. If SQLINSTANCE $ALL is used, then it must appear in the first operation of the batch file. Each operation in the batch file is performed for all SQL Server instances on the client where the batch file is executed. Also, it is not necessary to specify an SQLHOST or SQLINSTANCE on any subsequent operations.
STANDBYPATH	string	no	none	Specify a fully- qualified file path to use for the standby redo log.
STOPAFTER	datetime string	no	none	Specifies datetime for RESTORETOMARK options. The datetime string is formatted as YYYY/MMDDHH:MM:SS.
STOPAT	datetime string	no	none	Specifies the point-in-time recovery of a transaction log. The datetime string is formatted as YYYY/MMDDHH:MM:SS.
STORAGEIMAGE	string	no	none	Used for restoring a database that was backed up using a grouped Snapshot Client snapshot. STORAGEIMAGE identifies the image with which the physical files are associated.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
STRIPES	integer	no	1	Number of stripes. Range is 1-32.
TO	file path	no	none	Specifies a filegroup destination path. Required for each MOVE keyword. Also must sequentially follow each MOVE entry. The value may be delimited with single quotes.
TRACELEVEL	MIN, MID, MAX	no	MIN	Trace level.
TRXOPTION	NOTRUNC, TAILLOG	no	none	SQL Server transaction log backup options. If NOTRUNC is not selected, then the transaction log can be backed up and truncated. If TAILLOG is selected, the tail log is backed up and restored.
USERID	string	no	sa	User ID for logging into SQL Server. This keyword is ignored if you use integrated security.
VDITIMEOUTSECONDS	integer	no	300	Timeout interval for SQL Server Virtual Device Interface
VERIFYONLY	TRUE, FALSE	no	FALSE	Tells SQL Server to verify a backup image but not to restore it.

Table 12-2 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
VERIFYOPTION	NONE, STOPONERROR CONTINUEAFTERERROR	no	NONE	This option is only valid for the databases that have an active page. STOPONERROR performs verification and stops if a verification error occurs. CONTINUEAFTERERROR performs verification but continues if a verification error occurs.

Creating a batch file

You can use any of the backup or restore dialog boxes to create a batch file that contains a NetBackup for SQL Server script. This script can be executed at a later time from the Manage Scripts dialog box.

Or you can launch the script from the `dbbackex` command line program or through the NetBackup scheduler. See the example batch files.

See "About sample backup batch files for legacy SQL Server policies" on page 252.

To create a batch file

1. Select **File > Backup SQL Server objects** or **File > Restore SQL Server objects**.
2. Select the object you want to back up or restore.
3. Select the backup or restore options.

   See "Options for SQL Server backup operations" on page 210.

   See "Options for NetBackup for SQL Server restores" on page 81.
4. In the **Backup script** or **Restore script** group, click **Save**.
5. Click **Backup** or **Restore**.
6. Specify the following folder for the batch file:

   install_path\NetBackup\DbExt\MsSql\ folder.

   Batch files must reside on the host from which they executed. If you perform actions on a remote host, the batch file must reside on that remote host.

   See "Registering authorized locations used by a NetBackup database script-based policy" on page 268.

7 Give the file a unique name with the extension `.bch`.

8 Click **Save**.

Alternatively, you can select the name of an existing file and NetBackup appends the new script to it.

9 Click **Yes** to open and edit the batch file.

See "About sample backup batch files for legacy SQL Server policies" on page 252.

Running batch files

Once you have created a batch file, you manually run it from the NetBackup for SQL Server interface.

To run a batch file

1 Log on to the host and instance you want to access.

See "Selecting the SQL Server host and instance" on page 209.

2 Select **File** > **Manage script files**.

3 Double-click the batch file.

4 Click **Start**.

5 To monitor the operation, select **File** > **View status**.

Adding a new SQL Server legacy policy

This topic describes how to create a SQL Server legacy policy that uses clients and batch files to perform backups.

Note: To perform multistreamed backups and restores, or if you have multiple network interfaces, you need to perform other configuration.

See "Configuring multistriped backups of SQL Server" on page 76.

See "About configuration of SQL Server backups with multiple NICs" on page 171.

To add a new SQL Server legacy policy

1 Log on to the master server as administrator (Windows) or root (UNIX).

2 Open the NetBackup Administration Console.

3 If your site has more than one master server, choose the one on which you want to add the policy.

4 In the left pane, expand **NetBackup Management** and select **Policies**.

5 Select **Actions > New > Policy**.

6 In the **Add a New Policy** dialog box, in the **Policy name** box, type a unique name for the new policy.

7 Click **OK**.

8 In the **Add New Policy** dialog box, in the **Policy type** list, select **MS-SQL-Server**.

The database agent policy type does not appear in the drop-down list unless your master server has a license for the database agent.

9 Complete the entries on the **Attributes** tab.

See "About policy attributes" on page 45.

10 On the **Instances and Databases** tab, select **Clients for use with batch files**.

The tab name changes to **Clients** and the **Backup Selections** tab now lets you specify and browse for scripts.

11 Add other policy information as follows:

- Add schedules.
  See "About schedule properties " on page 202.
- Add clients.
  See "Adding clients to a policy" on page 207.
- Add batch files to the backup selections list.
  See "Adding batch files to the backup selections list " on page 208.

12 When you have added all the schedules, clients, and backup selections you need, click **OK**.

About schedule properties

Each policy has its own set of schedules. These schedules initiate automatic backups and specify when a user can initiate operations. Some schedule properties that have a different meaning for database backups than for file system backups. Other schedule properties vary according to your specific backup strategy and system configuration. See the NetBackup Administrator's Guide, Volume I.

Table 12-3 Description of schedule properties

Property	Description
Type of backup	Specifies the type of backup that this schedule can control. The selection list shows only the backup types that apply to the policy you want to configure. See "Legacy policy backup types" on page 203.
Schedule type	You can schedule an automatic backup in one of the following ways: ■ **Frequency** **Frequency** specifies the period of time that can elapse until the next backup operation begins on this schedule. For example, assume that the frequency is 7 days and a successful backup occurs on Wednesday. The next full backup does not occur until the following Wednesday. Typically, incremental backups have a shorter frequency than full backups. ■ **Calendar** The **Calendar** option lets you schedule the backup operations that are based on specific dates, recurring week days, or recurring days of the month.
Multiple copies	If you want to specify multiple copies of a backup for the policy, configure **Multiple copies** on the application backup schedule. If using Snapshot Client, also specify **Multiple copies** on the automatic schedule.

Legacy policy backup types

Table 12-4 shows that the backup types you can specify for a NetBackup for SQL Server legacy policy that uses clients and batch files. Intelligent Policies have a different set of backup types.

Table 12-4 Legacy policy backup types

Backup type	Description
Application Backup	The application backup schedule enables user-controlled NetBackup operations from the client. These operations include those initiated from the client and those initiated by a full schedule on the master server. NetBackup uses the application backup schedule when the user starts a backup manually. Configure at least one application backup schedule for each database policy. The Default-Application-Backup schedule is configured automatically as an application backup schedule.

Table 12-4 Legacy policy backup types *(continued)*

Backup type	Description
Full Backup	This schedule specifies the dates and times for NetBackup to automatically start backups as indicated in the batch file (full, differential, or transaction log). NetBackup runs the batch files in the order that they appear in the file list. If there is more than one client in the policy, the batch files are run on each client. See "Keywords and values used in batch files" on page 189. See "Converting differential backups to full backups" on page 204.

Converting differential backups to full backups

If a differential backup runs and a full backup does not already exist for the database or filegroup, NetBackup can convert the backup to a full backup. Similarly, NetBackup can convert transaction log backups if a full backup does not already exist for the database. Enable this behavior with the keyword `CONVERTBACKUP`.

See "Keywords and values used in batch files" on page 189.

NetBackup only converts a differential backup if a full backup was never performed on the database or filegroup. If a full backup does not exist in the NetBackup catalog but SQL Server detects an existing full LSN, NetBackup performs a differential backup and not a full. In this situation, you can restore the full backup with native tools and any differentials with the NetBackup MS SQL Client. Or, if NetBackup expired the backup, you can import the full backups into the NetBackup catalog. Then you can restore both the full and the differential backups with the NetBackup MS SQL Client.

The agent checks to determine if a full backup was ever performed for each database. If no previous full backup exists, the backup is converted to a full as follows:

- If you select a database for backup, the backup is converted to a full database backup.
  If you select **Read-write filegroups** for the **Type of Backup**, the backup is converted to a full read/write filegroup backup.
- If you select a filegroup for backup, NetBackup does the following:
  - If the filegroup is the default database filegroup, NetBackup converts the backup to a full filegroup backup.
  - If the filegroup is a secondary filegroup and a backup of the primary filegroup does not exist, NetBackup converts the backup to a partial full database backup. This backup contains the selected filegroup and default filegroup.

- If the filegroup is a secondary filegroup and a backup of the primary filegroup does exist, NetBackup converts the backup to a full filegroup backup of the selected filegroup.

- If you perform a partial differential backup, NetBackup does the following:
  - If no previous full backup exists for the default filegroup, NetBackup adds the filegroup to the backup and converts the operation to a full partial backup.
  - If a previous full backup exists for the default filegroup but a secondary filegroup in the files list does not have a full backup, NetBackup converts the operation to a full partial backup.
- The `CONVERTBACKUP` option also detects if a full recovery database was switched to the simple recovery model and back to the full recovery model. In this scenario, the log chain is broken and SQL Server requires a differential backup before a subsequent log backup can be created. If NetBackup detects this situation, the backup is converted to a differential database backup.

Configuring an application backup schedule

A database backup requires an application backup schedule. You cannot perform backups if this type of schedule is not included in the policy. The NetBackup for SQL Server agent automatically creates this schedule and names it **Default-Application-Backup**.

The backup window for an application backup schedule must encompass the time period during which all scheduled jobs and client-initiated jobs can occur. This window is necessary because the application backup schedule accepts the backup request from NetBackup for SQL Server regardless of whether the backup was initiated from an automatic schedule or from the client. You can choose to set the window for the application backup schedule for 24 hours per day, seven days per week. This window ensures that your operations are never locked out due to the application backup schedule.

For any policies that include read-only filegroups, consider creating a schedule with a retention level set to infinity. This level can enable you to avoid redundant backups.

To configure an application backup schedule

1. In the **Policy** dialog box, click the **Schedules** tab.

   To access the **Policy** dialog box, double-click the policy name in the **Policies** list in the NetBackup Administration Console.
2. Double-click the schedule that is named **Default-Application-Backup**.
3. Specify the other properties for the schedule.

   See “About schedule properties ” on page 202.

Example application backup schedule

Assume the following:

- Users perform database backup operations during business hours, 08:00 to 13:00.
- The automatic backups that use this policy start between 18:00 and 22:00.

In this scenario, the application backup schedule must have a start time of 0800 and a duration of 14 hours. Alternatively, the schedule can have two windows each day; one with a start time of 0800 and duration of 5 hours, and another with a start time of 1800 and a duration of 4 hours.

Table 12-5 Example settings for a NetBackup for SQL Server application backup schedule

Schedule option	Setting
Retention	2 weeks
Backup window	Sunday through Saturday 00:08:00 - 22:00:00

Configuring automatic backup schedules

If you put multiple batch files in the same policy, they run during each automatic backup session for that policy. You may have a variety of SQL Server backup operations that you want to run on different schedules. In this case, you may want to create multiple policies each with an automatic backup schedule that is different. Then assign each batch file to the policy that uses the appropriate automatic backup schedule.

If you plan to have NetBackup perform automatic backups, or if you use Snapshot Client features, you need one or more automatic backup schedules.

To configure an automatic backup schedule

1. On the **Policy** dialog box, click the **Schedules** tab.
2. Click **New**.
3. Specify a unique name for the schedule.
4. Select the **Full Backup** schedule.

   See “Legacy policy backup types” on page 203.

5 Specify the other properties for the schedule.

See “About schedule properties ” on page 202.

6 Click **OK**.

Example automatic backup schedule

Table 12-6 shows example settings for an automatic backup schedule.

Table 12-6 Example settings for a NetBackup for SQL Server automatic backup schedule

Schedule property	Setting
Retention	2 weeks
Frequency	Every week
Backup window	Sunday, 18:00:00 - 22:00:00

Adding clients to a policy

The client list is the list of hosts on which your batch files are run during an automatic backup. A NetBackup client must be in at least one policy but can be in more than one.

For a NetBackup for SQL Server policy, clients you want to add must have the following items installed or available:

- SQL Server
- NetBackup client or server
- The backup or restore batch files

Note: Each batch file must be present on each client.

To add clients to a NetBackup for SQL Server policy

1 Open the policy you want to edit or create a new policy.

To access the **Policy** dialog box, double-click the policy name in the **Policies** list in the NetBackup Administration Console.

2 Before you can add clients, you must select **Clients for use with batch files** on the **Instances and Databases** tab.

3 Click the **Clients** tab.

4 Click **New**.

5 Type the name of the client and select the hardware and operating system of the client.

If SQL Server is installed in a cluster, specify the virtual name of the SQL Server as the client name.

Note: For NetBackup 8.1, if you installed NetBackup on more than one node in the SQL Server cluster, you must perform additional configuration.

See "Reviewing the auto-discovered mappings in Host Management" on page 66.

See "Configuring mappings for restores of a distributed application, cluster, or virtual machine " on page 64.

---

6 Choose one of the following:

- To add another client, click **Add**.
- If this client is the last client you want to add, click **OK**.

7 In the **Policy** dialog box, click **OK**.

Adding batch files to the backup selections list

The backup selections list in a database policy has a different meaning than for non-database policies. For example, in a Standard or Microsoft Windows policy, the list contains files and directories to be backed up. In a database policy, you can specify batch files to run. (For NetBackup for SQL Server, the scripts are called batch files and have the `.bch` extension.) Batch files describe the backup operations you want to start. You can start them by initiating manual or scheduled operations from the NetBackup server. These files reside on the client and direct the operation of NetBackup for SQL Server and SQL Server.

Add batch files if you want a policy that runs scheduled backups. All batch files that are listed in the backup selections list are run for manual backups and for automatic backup schedules. Create the schedules on the **Schedules** tab. NetBackup runs the batch files in the order that the batch files appear in the backup selections list.

Note: Specify the correct batch file names in the backup selections list to prevent an error or possibly a wrong operation.

To add batch files to the backup selections list

1. Ensure that the batch file resides on the client.

   See "Registering authorized locations used by a NetBackup database script-based policy" on page 268.

2. Open the policy you want to edit or create a new policy.
3. Before you can add batch files, you must do the following:
   - On the **Instances and Databases** tab, select **Clients for use with batch files**.
   - On the **Clients** tab, add one or more clients.
4. Click the **Backup Selections** tab.
5. Click **New**.
6. In the **Add Backup Selection** dialog box, specify the names of the batch files that you want to use. Specify the file name in one of the following ways:
   - Click **Browse**. Navigate to and select the batch file, then click **OK**.
   - In the **Script** box, type the full path name of a batch file on the client, then click **Add**.
     For example:

     ***install_path*\NetBackup\DbExt\Mssql\bkup.bch**

     You must indicate the full pathname of the batch file.
7. Add any other batch files.
8. Click **OK** to add the batch files to the backup selections list.
9. Click **OK**.

Selecting the SQL Server host and instance

Use this procedure to set which SQL Server host and the instance that you want the NetBackup MS SQL Client to access. The user ID and password are only required if the host uses standard or mixed security. If applicable, you only need to provide these credentials when you first open the NetBackup MS SQL Client.

To select the SQL Server host and instance

1. Open the NetBackup MS SQL Client.
2. Select **File > Set SQL Server connection properties**.

3 In the **SQL Server connection properties** dialog box, from the **Host** drop-down list, select the SQL Server host.

You can type a host name if it does not appear in the drop-down list. If you select a remote host and click **Apply**, the **Host type** is shown as "remote".

4 From the **Instance** drop-down list, select the SQL Server instance.

You can type an instance name if it does not appear in the drop-down list. You can designate the default instance either by setting the Instance box to <default> or to empty (no spaces).

5 Click **Apply** to save your changes.

6 Click **Close**.

Options for SQL Server backup operations

Table 12-7 describes the options that are available when you perform backups. These options appear in the **Backup Microsoft SQL Server Objects** dialog box after you select **File** > **Backup SQL Server objects**.

Caution: Do not enable multiplexing if the policy is also configured with multiple stripes. Restores fail when both multiplexing and multiple stripes are configured for a backup policy.

Table 12-7 Options for SQL Server backup operations

Option	Description
Expand database	This pane lets you traverse live databases. You can expand the SQL Server instance to view its databases. Expand each database to view its filegroups or expand a filegroup to view its files. You can select any object in this pane to view its constituent objects in the right-hand pane.
Select database(s) for backup from *instance* *host\instance*	Select the objects that you want to back up from this pane. This pane displays the list of constituent database objects of the selected host and instance in the left-hand pane. You can select one or more objects (databases) in this pane.

Table 12-7 Options for SQL Server backup operations *(continued)*

Option	Description
Type of Backup	The following backup types are available: ■ **Full** Create a full database backup. ■ **Full differential** Create a differential backup. ■ **transaction log** Create a transaction log backup. This type of backup is only available for databases. When you select this type of backup, you then need to select a backup option from the **Transaction log backup options** list. ■ **Read/write filegroups** Create a backup of read-write filegroups in a database. ■ **Differential on read/write filegroups** Create a differential backup of read-write filegroups in a database. ■ **Create a template for partial backup** Create a backup of only the selected filegroups in a database. ■ **Create a template for partial differential backup** Create a differential backup of only the selected filegroups in a database.
Transaction log backup options	The following options are available when you have chosen a transaction log backup type: ■ **Back up and truncate transaction log** Back up the transaction log and remove the inactive part of the transaction log. ■ **Back up transaction log, but do not truncate it** Back up a transaction log without truncating it. ■ **Back up and restore tail log** Back up and recover the tail log from disk.
Use SQL compression	Select this option if you want to use SQL Server to compress the backup image. If you enable SQL Server compression, do not enable NetBackup compression.
Backup script	■ **Launch Immediately** Start the backup operation immediately. **Launch immediately** is disabled if you are logged into a SQL Server instance that is not on the local host. If you generate a script for a non-local host, then it must be executed on that host. ■ **Save** Generate a script that can be started at a later time.

Table 12-7 Options for SQL Server backup operations *(continued)*

Option	Description
Back up	In the right-hand pane, choose one of the following backup options: ■ **Selected** Back up only the objects selected. ■ **All but selected** Back up all of the objects, except those selected. ■ **All** Back up all of the objects.
Stripes	Set the number of backup stripes that you want SQL Server to create for your backup. Type a number from 1 to 32. **Caution:** Do not enable multiplexing if the policy is also configured with multiple stripes. Restores fail when both multiplexing and multiple stripes are configured for a backup policy. See "Configuring multistriped backups of SQL Server" on page 76.
Resume options for this selection	■ **Do not resume unsuccessful backups** ■ **Retry from the beginning** Restart failed backups after waiting 60 seconds. ■ **Save work and restart at point of failure** Divide the backup into multiple streams and back up separately. Any streams that fail are restarted after 60 seconds. This option is available when the following conditions are met: ■ Exactly one object has been selected, ■ The object that is selected for backup is a database or filegroup and the backup type is full, ■ The SQL Server object uses the "full" or "bulk-logged" recovery method.
NetBackup policy	If this host is the NetBackup master server, then this list includes all active policies of type MS-SQL-Server. You can select one of these policies or type the name of a policy. The default is <any>. If you select the default, then NetBackup selects which MS-SQL-Server policy to use.
Page verification	This option is enabled for objects have a page verification type that is either torn page detection or checksum. All of the objects in the right-hand pane must have the proper verification type. This indicates a performance penalty when you use page verification. ■ **Do not perform verification** Do not perform page verification before you run the backup. ■ **Perform verification** Perform page verification when you run the backup and stop the backup if a verification error is encountered.

Table 12-7 Options for SQL Server backup operations *(continued)*

Option	Description
Backup	Start a database backup or generate a database backup script. This option is enabled only when you select an object to back up.

About viewing the properties of the objects selected for backup

You can view the properties of any object in the **Backup Microsoft SQL Server Objects** dialog box by right-clicking the object. Table 12-8 describes the properties of objects that are selected for backup.

To view the properties of an object that is selected for backup

1. Select **File > Backup SQL Server objects**.
2. In the **Backup Microsof SQL Server Objects** dialog box, in the right pane, right-click an object and select **Properties**.
3. When you finish, click **OK**.

Table 12-8 Properties of the objects that are selected for backup

Property	Description
Object type	Database, database filegroup, database file, or transaction log.
Object name	Name of the object.
Parent (database, instance, filegroup, etc.)	Name of the object's parent.
SQL Server instance	SQL Server instance the object belongs to.
File size	The size of the component files. This size should closely match the size of a backup snapshot.
Data size	Size of the backup stream. Applies to databases only.
Page verification	The type of SQL Server page verification that is configured for selected databases, filegroups, and logical files. The available values are: none, torn page detection, or checksum.
Read-only/read-write	The attribute that is applied to the filegroup.
On-line/off-line	The status of the filegroup.

Table 12-8 Properties of the objects that are selected for backup *(continued)*

Property	Description
Path	(Database files only) The absolute path of the database file.

Performing user-directed backups of SQL Server databases

This procedure describes how to perform a database backup.

To perform a user-directed backup of a SQL Server database

1. Open the NetBackup MS SQL Client interface.
2. Select the host and instance you want to access.

   See "Selecting the SQL Server host and instance" on page 209.
3. Select File > **Backup SQL Server objects**.
4. In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, select the database instance.
5. In the right pane, select one or more databases that you want to back up.
6. Select the **Type of Backup**.

   Select one of the following:

   - To perform a full backup, select **Full Backup**.
   - To back up the database with the differential option, select **Perform differential backup**.
7. Select the backup options.

   See "Options for SQL Server backup operations" on page 210.
8. Click **Backup**.
9. When you are prompted to start the backup, click **Yes**.
10. To view the progress of the backup, select File > **View status**.

Backing up SQL Server transaction logs

This procedure describes how to perform a transaction log backup.

Caution: Ensure that the entire sequence of transaction logs generated following any database backup are maintained on the same NetBackup server. Back up all transaction logs to the same facility and do not allow any logs to expire before the others.

To back up a transaction log

1. In SQL Server, set the **Recovery Model** setting to either **Full** or **Bulk-logged**.
2. Open the NetBackup MS SQL Client interface.
3. Select the host and instance you want to access.

   See "Selecting the SQL Server host and instance" on page 209.
4. Select **File > Backup SQL Server Objects**.
5. In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, select the database instance.
6. In the right pane, select one or more databases whose transaction logs you want to back up.
7. In the **Type of Backup** list, select **transaction log**.
8. From the drop-down list, select the transaction log option. For more information, see the following table.

Back up and truncate transaction log	Back up the transaction log and remove the inactive part of the transaction log.
Truncate transaction log, but don't back it up	Truncate the log without performing a backup.
Back up and restore tail log	Back up and recover the tail log from disk.

9. Select the backup options.
10. Click **Backup**.

    To view the progress of the backup, select File > **View status**.

Backing up SQL Server database filegroups

More information is available on how to use read-write and read-only filegroups in your backup strategy.

See "Backing up read-write filegroups" on page 217.

See "Backing up read-only filegroups" on page 216.

To back up a database filegroup

1. Open the NetBackup MS SQL Client interface.
2. Select the host and instance you want to access.

   See "Selecting the SQL Server host and instance" on page 209.
3. Select File > **Backup SQL Server objects**.
4. In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, expand the instance name.
5. Select a database whose filegroups you want to back up.
6. In the right pane, select one or more filegroups that you want to back up.
7. Select the backup options.
8. Click **Backup**.

   To view the progress of the backup, select File > **View status**.

Backing up read-only filegroups

When you separate read-only and read-write filegroups in your backup strategy, you can reduce total media usage and the total time you spend on backup operations. To back up read-only filegroups you must first create a separate policy for this type of backup. You can also verify that all read-only filegroups are backed up.

See "Reducing backup size and time by using read-only filegroups" on page 164.

See "Viewing SQL Server read-only backup sets" on page 217.

To back up read-only filegroups

1. Open the NetBackup MS SQL Client interface.
2. Create a batch file that includes the read-only filegroups.

   All read-only filegroups must be included in some combination of full, or individual filegroup and file backups. You only need to perform this backup one time.
3. In the NetBackup Administration Console, create a backup policy for read-only filegroups.
   - In the Application Backup schedule, set the **Retention** level of **Infinite**.
   - Add the batch file that you created to the backup selections list.

4 Back up the read-only filegroups.

5 If necessary, confirm all read-only groups are backed up by viewing the read-only backup set.

See “Viewing SQL Server read-only backup sets” on page 217.

Viewing SQL Server read-only backup sets

If you perform periodic backups only on read-write filegroups, you can verify if you have retained backups of the read-only filegroups.

To view read-only backup sets

1 Open the NetBackup MS SQL Client interface.

2 Browse for the backup images that contain the read-only backup sets.

See “Browsing for SQL Server backup images” on page 80.

3 In the **Restore Microsoft SQL Server Objects** dialog box, expand the instance name.

4 Right-click the database and select **Properties**.

5 Click the "Read-only backup set" tab.

If the database does not contain read-only filegroups, then the message "This database does not contain any read-only filegroups." is shown. If backups do not exist for all of the read-only filegroups, then a list of the filegroups that were not backed up is shown. Finally, if a backup is found of all of the read-only filegroups, then the name appears of the latest image that contains this backup.

6 If there are any read-only filegroups that are not backed up, back them up as soon as possible. These backups ensure you can perform a full recovery.

7 Click **OK**.

Backing up read-write filegroups

When you separate read-only and read-write filegroups in your backup strategy, you can reduce total media usage and the total time you spend on backup operations. More information is available on backing up read-only filegroups.

See “Backing up read-write filegroups” on page 217.

See “Backing up read-only filegroups” on page 216.

Note: Immediately back up any filegroup when you change it from read-write to read-only.

To back up read-write filegroups

1. Open the NetBackup MS SQL Client interface.
2. Select **File > Backup SQL Server objects**.
3. In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, select the database instance.
4. In the right pane, select one or more databases that you want to back up.
5. Select the **Type of Backup**, as follows:
   - To perform a full backup of the read-write filegroups, select **Read-write filegroups**.
   - To perform a differential backup of the read-write filegroups, select **Differential on read-write filegroups**.
6. Select the backup options.
7. From the **Backup script** group, select **Save**.
8. Click **Backup**.

   Note the location where the batch file is saved. This batch file is added to the policy that backs up the read-write filegroups.
9. Open the NetBackup Administration Console.
10. Create a backup policy for read-write filegroups.
    - Create a **Full Backup** schedule with the wanted retention period.
    - Add the batch file that you created to the backup selections list.
11. (Optional) Manually back up the read-write filegroups.

    If you do not perform a manual backup at this time, the backup runs automatically through the schedule you created in step 10.

Backing up SQL Server database files

This procedure describes how to back up database files.

To back up a database file

1. Open the NetBackup MS SQL Client interface.
2. Select the host and instance you want to access.

   See “Selecting the SQL Server host and instance” on page 209.
3. Select File > **Backup SQL Server objects**.

4 In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, expand the instance name and database.

5 In the left pane, select the filegroup that contains the files you want to back up.

6 In the right pane, select one or more files that you want to back up.

7 Select the backup options.

8 Click **Backup**.

To view the progress of the backup, select File > **View status**.

Performing partial database backups

This procedure describes how to create a script for to perform a partial database backup.

To perform a partial database backup

1 Open the NetBackup MS SQL Client interface.

2 Select the host and instance you want to access.

See "Selecting the SQL Server host and instance" on page 209.

3 Select **File > Backup SQL Server objects**.

4 In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, select the database instance.

5 In the right pane, select a database that you want to back up.

6 For the **Type of Backup**, select one of the following:

- **Create a template for partial backup**.
- **Create a template for partial differential backup**.

7 Select the backup options.

8 Click **Backup**.

9 In the **Save Script As** dialog box, specify a file name and click **OK**.

10 When you are prompted to open the template, click **Yes**.

11 Edit the template by uncommenting the filegroups that you want to include in the backup. You must uncomment at least one filegroup.

For example, replace:

```
#
#  If you wish to include filegroup DBA_FG1 in the partial backup,
#  then remove the hash mark that precedes the following line.
#FILEGROUP DBA_FG1
```

with:

```
#
#  If you wish to include filegroup DBA_FG1 in the partial backup,
#  then remove the hash mark that precedes the following line.
FILEGROUP DBA_FG1
```

12 When you are finished modifying the template, save it.

13 To run the backup, select File > **Manage script files**, select the script you created, and click **Start**.

Performing a backup of a remote SQL Server installation

You can use NetBackup for SQL Server to back up databases on a remote host. Generated batch files must be saved on the remote host. You can launch the operation from the local installation of NetBackup for SQL Server, from an automatic backup policy, or from a manual backup.

To perform a backup of a remote SQL Server installation

1 Select the host and instance you want to access.

See "Selecting the SQL Server host and instance" on page 209.

2 Select **File > Backup SQL Server objects**.

3 Select the options for the operation.

See "Options for SQL Server backup operations" on page 210.

Save is enabled in the backup dialog box. **Launch immediately** is disabled because the generated script must be executed on the remote host that you are logged on to.

4 Click **Backup**.

5. In the **Save Script As** dialog box, navigate to the *install_path*\NetBackup\DbExt\MsSql\ folder on the remote host, and save the batch file there.
6. Launch the backup operation.

   Do one of the following:

   - Run the operation from the local installation of NetBackup for SQL Server.
   - Create a new policy that includes the remote SQL Server client. Add the batch file to the **Backup Selections** list in the policy.

About file checkpointing with NetBackup for SQL Server

Use file checkpointing if you need to perform a large backup and want to save completed work in case the operation fails before it completes. When file checkpointing is enabled, the database or filegroup is divided into file sets and backed up as separate units. The following batch file command initiates file checkpointing:

MAXRESTARTSETS *integer*

The backup operation is split into the number of operations equal to the *integer* value. If the number of files is less than the *integer* value, then the number of separate operations is equal to the number of files.

File checkpointing is available for databases and filegroups that are backed up as streams or with the snapshot option. However, the following restrictions exist:

- The backup object must contain at least two files.
- The recovery model of the database cannot be “simple”.
- If the snapshot option is used for backup, then the method cannot be Instant Recovery. However, file checkpointing that uses Instant Recovery to a storage unit is supported.
- The batch file that you use for a file checkpoint backup can specify only one database or filegroup. You cannot use the `DATABASE $ALL` option.

When you use file checkpointing for backing up a full database, NetBackup for SQL Server automatically splits the database into fileset components. Recovering the database from components requires a restore of the transaction log. NetBackup for SQL Server automatically includes a backup log directive in the generated batch file when you choose file checkpointing from the backup dialog box.

About automatic retry of unsuccessful SQL Server backups

NetBackup for SQL Server provides the following options to retry unsuccessful backup attempts.

Automatic retry	NetBackup for SQL Server keeps track of the unsuccessful backups that may have resulted from the execution of a batch file. When the initial backup attempt is complete, the agent rewrites the batch file, including only those operations that failed. The rewritten batch file is launched automatically.
Manual retry	A manual retry is similar to an automatic retry except that NetBackup does not launch the rewritten batch file. Instead it is written to the *install_path*\dbext\mssql\temp directory. The user can then choose when to run the new batch file.

To use automatic retry, add the following line to your batch file.

```
RESTARTTYPE AUTO
```

By default, the unsuccessful backups are retried one time automatically after 60 seconds. To change the delay following the unsuccessful attempt, then add the following to your batch file.

```
RESTARTWAITSECONDS <integer>
```

You can also specify the number of retries. Add the following to your batch file.

```
NUMRESTARTS <1 to 9>
```

To use manual retry, add the following line to your batch file.

```
RESTARTTYPE MANUAL
```

Retry may also be used with file checkpoints. Any parts of the operation that fail can be written to a new batch file that can be launched either automatically or manually.

See “About file checkpointing with NetBackup for SQL Server ” on page 221.

You can enable file checkpointing with automatic retry in the backup dialog in the NetBackup for SQL Server Client. Select a single database (or filegroup), then from the **Resume options for this selection** list, select **Save work and restart at point of failure**.

This action creates a batch file that contains the following scripting:

```
MAXRESTARTSETS 32
RESTARTWAITSECONDS 60
NUMRESTARTS 1
```

`MAXRESTARTSETS 32` means that up to 32 pieces are backed up independently. The keywords `RESTARTWAITSECONDS` and `NUMRESTARTS` are synonymous with the following:

```
RETRYWAITSECONDS 60
NUMRETRIES 1
```

These keywords indicates the following things: first, that an automatic retry is launched after 60 seconds for all of the pieces that failed to get backed up on the first time. Second, the restart is attempted only one time. You can manually change either of these parameters.

In addition, you can choose to not have the retry script automatically launched. Replace the `NUMRETRIES` command with `RETRYTYPE MANUAL`. For example, replace the following:

```
NUMRETRIES 1
```

with

```
RETRYTYPE MANUAL
```

Note: All of the keyword-value pairs that are described in this topic are only permitted in the first operation of the batch file.

Chapter 13

Performing user-directed operations with dbbackex

This chapter includes the following topics:

Using dbbackex to perform user-directed operations for SQL Server

`dbbackex` is a command line interface program you can use to perform backups and restores of SQL Server. To start `dbbackex`, run the following from a command prompt:

```
install_path\NetBackup\bin\dbbackex -f file [-p policy][-u userid][-pw password]
[-s server][-np]
```

Refer to the description of the following parameters.

file — The name of the batch file, which describes the operations you want to start.

See "Running batch files" on page 201.

policy	The MS-SQL-Server policy type NetBackup uses for the operations that are specified in the batch file. This parameter is ignored for restore operations. The NetBackup server can retrieve the dump file based entirely on the image names that are specified in the batch file for each restore. The policy name is used for databases backups. If it is omitted, then the NetBackup server uses the first active SQL Server policy that it finds in its policy list. This policy name is used for all of the backup operations that are specified in the batch file.
userid	is the SQL Server user ID for logging into the database management system.
password	is the SQL Server password for logging into the database management system.
server	is the name of the host for the NetBackup master server that you want to back up to or restore from. If this parameter is omitted, then the client uses the default server according to the Windows NetBackup client configuration. See the NetBackup Backup, Archive, and Restore Getting Started Guide for more information.
-np	tells `dbbackex` not to create a message box to indicate the operation status when it has completed. Otherwise, a message appears when `dbbackex` completes. That message tells you how many operations in the batch file were successful and how many failed.

Note: Any of the options can be delimited with double quotation marks. For example, use delimiters if the file name contains spaces.

Note: To protect logon passwords for SQL Server, do not use the `-u` or `-pw` parameters. By omitting these parameters, you can force NetBackup for SQL Server to read the default SQL Server logon data from an encrypted file.

See "Starting the NetBackup MS SQL Client for the first time" on page 79.

Using client-based schedulers with dbbackex

`dbbackex` lets you employ your choice of client-based schedulers to automatically initiate NetBackup for SQL Server operations.

The following schedulers are available:

- The Windows Task Scheduler. Instructions for using this scheduler are provided in the Microsoft Windows online documentation.
- The SQL Server Scheduler. This scheduler is closely integrated with SQL Server. It can be accessed through the Microsoft SQL Server Enterprise Manager.

One distinct advantage of the SQL Server Scheduler is that you can create scripts for database maintenance operations. These operations are initiated as a result of database events that you define. For example, you can create a script that initiates `dbbackex` and tells it to back up a particular transaction log. You can also create an alert which invokes that script when the transaction log for this database becomes full.

Note: If you use `dbbackex` through a client-based scheduler, specify the `-np` option to ensure that a message box is not generated. However, before you use the scheduler with `dbbackex` consider the following. Try the `dbbackex` syntax on the console *without* the `-np` option. This command tests for the successful completion of the batch file that you have created for your operation.

Chapter 14

Using bplist to retrieve a list of SQL Server backups

This chapter includes the following topics:

About using bplist to retrieve SQL Server backups

You can use the `bplist` command to obtain restore images. Use this command if you plan to manually create a restore script, rather than through the NetBackup for SQL Server interface. See the NetBackup Commands Reference Guide for complete information about `bplist`.

To extract all of the NetBackup for SQL Server backups from a specific server for a specific client, run the following command from the Windows command prompt.

install_path\NetBackup\bin\bplist -C *client* -t 15 -S *server* -R \

where *client* is the host machine on which NetBackup for SQL Server resides and *server* is the host machine of NetBackup server.

The following example shows how to obtain the list of SQL Server backups that were backed up from client juneberry to server Cole:

```
C:\Program Files\NetBackup\bin\bplist -C juneberry -t 15 -S cole -R \
juneberry.MSSQL7.JUNEBERRY.db.pubs.~.7.001of003.20140920101716..C:\
juneberry.MSSQL7.JUNEBERRY.db.pubs.~.7.002of003.20140920101716..C:\
juneberry.MSSQL7.JUNEBERRY.db.pubs.~.7.003of003.20140920101716..C:\
juneberry.MSSQL7.JUNEBERRY.fil.pubs.pubsnew.7.001of001.20140919175149..C:\
juneberry.MSSQL7.JUNEBERRY\NEWINSTANCE.trx.abc.~.7.001of001.20140902170920..C:\
```

```
juneberry.MSSQL7.JUNEBERRY\NEWINSTANCE.fg.abc.PRIMARY.7.001of001.20140902170824.C:\
juneberry.MSSQL7.JUNEBERRY\NEWINSTANCE.db.Howard's
Barbeque.~.7.001of001.20140901085255..C:\
juneberry.MSSQL7.JUNEBERRY\NEWINSTANCE.inc.Howard's
Barbeque.~.7.001of001.20140903108552..C:\
juneberry.MSSQL7.COLE.db.pubs.~.7.001of001.20140907100101..C:\
juneberry.MSSQL7.COLE.db.pubs.~.7.001of001.20140908200234..C:\
```

Note: The colon and backslash that terminate each line are not part of the backup name.

About NetBackup for SQL Server backup names

The backup name is a string that consists of the following components. These components are separated by a delimiter that is specified by the character that precedes the “C” at the end of the backup image name.

Figure 14-1 Backup image name for a database filegroup

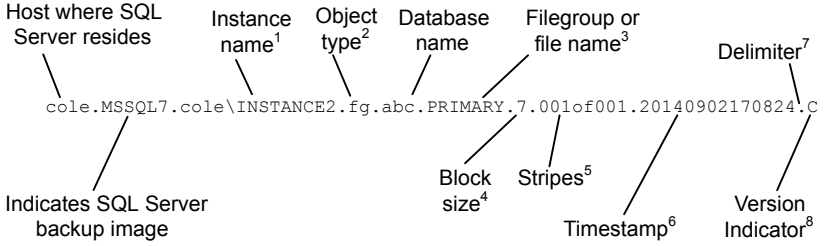
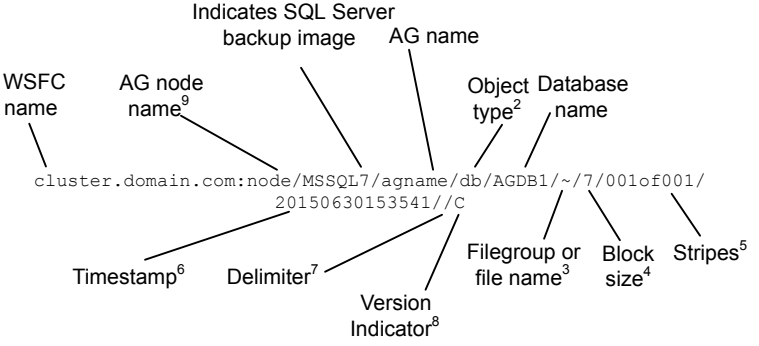


Figure 14-2 Backup image name for availability group (AG) database



1 - Named instances are formatted as <*host*>\<*instance-name*>. The default instance is the name of the host machine.

2 - The object types are as follows:

db	database
inc	database differential
trx	transaction log
fg	filegroup
fgd	filegroup differential
fil	file

3 - The name of the file or filegroup if the object type is a file or filegroup; otherwise the symbol ~ is used.

4 - The block size.

5 - Stripes are specified as <*stripe number*>of<*total stripes*>. non-striped backups are always `001of001`. For striped backups, <*total stripes*> is the total number of stripes for the backup. <*stripe number*> is the count number of the backup for that backup, starting with 001.

6 - The format of the timestamp is *YYYYMMDDHHMMSS*. The timestamp for AG backup images reflects Coordinated Universal Time (UTC). For non-AG backup images, the timestamp reflects the time zone that is configured for the NetBackup server.

7 - The delimiter, which immediately precedes the version indicator. For non-AG database images, this character is a period (.) by default. For AG database images, the character is a forward slash (/). However, if a period or slash is used in any of the fields, the delimiter may be another character.

8 - “C” is applied to all SQL Server backup image names, regardless of the NetBackup version.

9 - Backup images for AG databases are formatted as <*WindowsServerFailoverCluster*>:<*nodename*>/MSSQL7/<*AGname*>.

Chapter 15

SQL Server backups and restores in an SAP environment (legacy SQL Server policies)

This chapter includes the following topics:

About SQL Server backups and restores in an SAP environment

Note: SQL Server in an SAP environment is not supported for SQL Server Intelligent Policy.

With NetBackup you can perform scheduled SAP backups, in accordance with a predefined backup strategy, or manual backups. These backups may not be planned and may be necessary in exceptional situations. The practices that are described here are based on the practices SAP recommends in SAP/MS SQL Server DBA in CCMS.

The NetBackup backup and restore procedures for the SAP R/3 database are identical to the NetBackup procedures with any other SQL Server database.

You can create scripts to perform full or differential backups of databases and backups of transaction logs. In addition to the database backups and restores, NetBackup also provides the capabilities to back up the SAP file systems.

Creating batch files for automatic backups in for SQL Server in an SAP environment

NetBackup for SQL Server uses batch files to initiate database backup and restore operations. A batch file must be created for database backups and for transaction log backups. These batch files must then be added to the backup selections list in the backup policies that you created.

Creating a batch file for database backups

This topic describes how to create a batch file for database backups.

To create a script for database backups

1. Open the NetBackup MS SQL Client.
2. Select File > **Backup SQL Server objects**.
3. In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, expand the database instance.
4. In the right pane, select the R/3 database.
5. From the **Type of Backup** list, select the type of backup you want to perform, **Full, or Full differential**.
6. Under **Backup Script**, select **Save**.
7. Click **Backup**.
8. Specify a file name and click **Save**.

   Alternatively, you can select the name of an existing file, and NetBackup appends the new script to it.
9. Click **Yes** to open and edit the batch file.

Creating a batch file for transaction log backups

This topic describes how to create a batch file for transaction log backups.

To create a batch file for transaction log backups

1. Before starting a transaction log backup, the database administrator should set the **Transaction log backup options** database option to off. This option on the SQL Server interface applies to the databases.

   The entire sequence of transaction logs generated following any database dump must be maintained on the same NetBackup server. NetBackup for SQL Server requires that you follow these guidelines in devising your backup strategy to ensure success in restoring your database.
2. Select File > **Backup SQL Server objects**.
3. In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, expand the database instance.
4. In the right pane, select the R/3 database.
5. For the **Type of Backup**, select **transaction log**.
6. Under **Backup Script**, select **Save**.
7. Click **Backup**.
8. Specify a file name and click **Save**.

   Alternatively, you can select the name of an existing file, and NetBackup appends the new script to it.
9. Click **Yes** to open and edit the batch file.

Monitoring backups on SQL Server

Check scheduled backups regularly to ensure that they completed successfully.

Always check the following:

- That the most recent backup has run successfully.
  See "About monitoring NetBackup for SQL Server operations" on page 237.
- All the backups in the backup cycle are executed according to the schedule. Gaps in a backup sequence can have serious consequences in a subsequent attempt to restore the database.

Restoring the R/3 database

This topic describes how to restore the R/3 database.

Determine how to perform the restore based on the following scenarios:

- If you have scheduled differential backups, review the information for that type of restore.

See "About including differential backups in a restore operation" on page 233.

- If the R/3 database disk system is damaged or the transaction log disk system is damaged, follow the instructions for that scenario.
  See "Restoring the R/3 database after a disk crash" on page 233.
- To perform a regular restore of the R/3 database, follow the instructions for that type of restore.
  See "Restoring the database backups and transaction log backups" on page 234.

About including differential backups in a restore operation

If you incorporated differential backups in the backup strategy, the restore process differs depending on the type of backups available.

Determine how to perform the restore based on which of the following differential backups you have:

- If differential backups were made after the last full database backup, restore the last database backup that is followed by the most recent differential backup. Then apply all subsequent transaction logs.
- If no differential backups were made since the last full database backup, restore the last full database backup and then apply all subsequent transaction logs.
- If several differential backups are available but the latest one cannot be read, restore the most recent full database backup. And restore the latest readable differential backup and apply all subsequently created transaction logs.

Restoring the R/3 database after a disk crash

This topic describes how to restore the database when the R/3 database disk system is damaged or the transaction log disk system is damaged. This process is only applicable to a configuration with three disk systems: one system for the R/3 database, one for the R/3 transaction logs and one for all others.

Note: The R3 database must not be in use when you are performing a restore operation. Make sure that all SAP services are stopped before you attempt a restore with NetBackup.

Warning: If the disk system on which the R/3 database resides is damaged, it is vital to immediately back up the currently active transaction log. This log backup is done to prevent loss of data. Without a backup of the current log, the database can only be restored to the status at the time of the last transaction log backup. If work has been carried out on the R/3 system since then, this work is lost.

To restore the R/3 database after a disk crash

1. Back up the current transaction log.
2. Replace damaged disks.

   Replacing damaged disks in a RAID disk system is normally a straightforward procedure. If you are uncertain how to proceed, see the documentation of your hardware vendor to learn how to handle the disks. The new disks must be formatted and assigned the same drive letter as the old disks.
3. Restore the database logs and transaction logs.

   The central phase of a restore operation is the reloading of the database backup and the application of the available transaction logs. When the database backup is reloaded, the database files are automatically recreated. The data is copied from the backup device to the newly created files. Once this copy has been done, the transaction logs are applied in the same sequence as they were originally made. In a final step, open transactions that were not completed at the time of the database failure are rolled back.

Restoring the database backups and transaction log backups

NetBackup MS-SQL server agent GUI provides for automatic staging. By selecting the latest transaction log backup, the GUI automatically restores the previous full database backup. It also restores any optional differential backups and subsequent transaction log backups. You can also use the option to specify a point in time to which to restore to.

Note: The R3 database must not be in use when performing a restore operation. Make sure that all SAP services are stopped before you attempt a restore with NetBackup.

Warning: To restore the R/3 database you first restore the most recent database backup and then the subsequent transaction logs. During the entire procedure, do not execute any transactions and do not shut down the database server. A server shutdown would write a checkpoint to the log and as a result you would not be able to restore further transaction logs.

To restore the database backups and transaction log backups

1. Restore the most recent database backup.
2. Restore the latest differential database backup (if available).

3 Restore all succeeding transaction log backups.

4 Restore the latest transaction log backup.

About manual backups of SQL Server in an SAP environment

The administrator on the master server can use the NetBackup Administration Console to manually execute an automatic backup schedule. This schedule can be for an "MS-SQL-Server" policy, where the R/3 database is specified in the backup script.

For more information, see the section on manual backups in the NetBackup Administrator's Guide, Volume I.

About policy configuration for SQL Server in an SAP environment

To automatically perform backups of an SAP environment, you need to create backup policies. A backup policy with the "MS-SQL-Server" policy type that is selected must be created for R/3 database backups. Batch files, which initiate the backup of the database and transaction logs, must be added to the backup selections list in the policy.

Information is available for how to create the batch files that are needed and how to configure backup policies.

For backups of the executables disk (a file-system backup), a backup policy must be created with the Windows policy type selected.

For information on Windows policies, see the NetBackup Administrator's Guide, Volume I.

Chapter 16

Troubleshooting

This chapter includes the following topics:

About monitoring NetBackup for SQL Server operations

Use the Activity Monitor in the NetBackup Administration Console to monitor NetBackup for SQL Server operations.

The agent also creates its own progress reports that you can view in the NetBackup MS SQL Client interface. Select **File > View status** to view the reports. The reports are saved in *install_path*\NetBackup\logs\user_ops\MsSql\logs.

Job details and progress reports include the following types of information:

- Summary information about the operation
- Information about the operation as it progresses
- Any error conditions or warnings that cause the operation to fail
- The final outcome of the operation, whether it succeeded or failed, and how long it took

The progress reports also provide additional details for operations, including the following:

- The SQL Server commands that NetBackup included in the batch file for operation.

```
OPERATION BACKUP
DATABASE "TestDB1"
OBJECTTYPE DATABASE
COPYONLY FALSE
BLOCKSIZE 7
MAXTRANSFERSIZE 6
NUMBUFS 2
STRIPES 1
SQLCOMPRESSION FALSE
VERIFYOPTION NONE
```

- The NetBackup server that performed the backup, the SQL Server instance and host you selected for the backup, and other policy information.

```
NBSERVER "servera"
SQLINSTANCE "SQL2K14"
SQLHOST "SERVERA"
POLICY "sql-server"
NBSCHED "full"
```

```
INF - Setting backup catalog name to: servera
```

- Progress of the backup or restore operation and any errors or failures that SQL Server encountered.

```
USER - Operation inhibited by NetBackup for Microsoft SQL
Server: Only a full or incremental database backup can be performed
on database <Archive> because it uses the simple recovery model or
has 'truncate log on checkpoint' set.

INF - OPERATION #1 of batch
C:\NBU\Veritas\NetBackup\dbext\mssql\temp\__01_35_42_508_00.bch
FAILED with STATUS 1 (0 is normal). Elapsed time = 6(6) seconds.

INF - Results of executing
<C:\NBU\Veritas\NetBackup\dbext\mssql\temp\__01_35_42_508_00.bch>:
<0> operations succeeded. <1> operations failed.

INF - The following object(s) were not backed up successfully.

INF - Archive
```

About NetBackup reports for SQL Server troubleshooting

The administrator has access to operational progress reports through administrator interfaces. Reports may be generated for following: Backup Status, Client Backups, Problems, All Log Entries, Media Lists, Media Contents, Images on Media, Media Logs, Media Summary, and Media Written. These reports may be generated for a specific time frame, client, or master server. See the NetBackup Administrator's Guide, Volume I for details.

About debug logging for SQL Server troubleshooting

The NetBackup master server and client software offers a comprehensive set of debug logs for troubleshooting the problems that can occur during NetBackup operations. Debug logging is also available for SQL backup and restore operations. After the cause of the problem is determined, you can disable debug logging.

You can control the amount of information that is written to debug logs.

See "Setting the debug level" on page 240.

For details on the contents of these debug logs, see the NetBackup Troubleshooting Guide.

For additional NetBackup client logs and NetBackup master server logs, see the online Help for the Backup, Archive, and Restore interface and the NetBackup Administrator's Guide, Volume I.

Creating all NetBackup debug logs for SQL Server troubleshooting

You can use the following procedure to create all NetBackup debug logs.

To create all debug logs

- Run the following batch file:

  install_path\NetBackup\logs\mklogdir.bat

See "About backup operation debug logging for SQL Server" on page 239.

See "About restore operation debug logging for SQL Server" on page 240.

About backup operation debug logging for SQL Server

After you perform a backup, debug logging information is placed in the *install_path*\NetBackup\logs directory. A subdirectory is created for each process. The debug log file is named ALL_ADMINS.mmddyy_0000x.log. For unified logging (VxUL), the log file is in a format that is standardized across Veritas products. For details on logging, see the NetBackup Troubleshooting Guide.

Client	Refer to the following logs: ■ bphdb (scheduled backups only) ■ dbclient ■ ncfnbcs (VxUL) ■ nbdisco (VxUL) ■ user_ops\mssql\logs
Master server	nbars (VxUL)
Snapshot backups	Refer to the following logs: ■ bpbkar (Snapshot Client) ■ nbfsd (Snapshot Client) ■ bppfi Instant Recovery

VMware backups	For ASC issues and failures, the following logs are created on the VM that is backed up: ■ `bpbkar` ■ `dbclient` ■ `ncfnbcs` (VxUL)

About restore operation debug logging for SQL Server

The following logs apply to restore operations.

Client	Refer to the following logs: ■ `bpbkar` (Snapshot Client) ■ `bpfis` (Snapshot Client) ■ `bppfi` (Instant Recovery) ■ `dbclient` ■ `user_ops\mssql\logs`
VMware restores from snapshots using Replication Director	See the Veritas VSS provider logs. See "Veritas VSS provider logs" on page 241.

Setting the debug level

To control the amount of information that is written to the debug logs, change the Database debug level. Typically, the default value of 0 is sufficient. However, technical support may ask you to set the value higher to analyze a problem.

The debug logs are located in *`install_path`*`\NetBackup\logs`.

Information is also available about the **Client Trace Level**. See "Setting the maximum trace level for NetBackup for SQL Server" on page 242.

To set the debug level

1. Open the **Backup, Archive, and Restore** interface.
2. Select **File > NetBackup Client Properties**.
3. Click the **Troubleshooting** tab.
4. Set the **General** debug level.
5. Set the **Verbose** debug level.
6. Set the **Database** debug level.
7. Click **OK** to save your changes.

Veritas VSS provider logs

The Veritas VSS provider records its activities in Windows Event Logs. Debug logs are also available at the following location:

install_path\Veritas VSS provider\logs

Enabling Veritas VSS provider logging in the registry

Enable the Veritas VSS provider logging on the NetBackup computer where SQL Server is installed.

To enable Veritas VSS provider logging in the registry

1. Log on as administrator on the computer where NetBackup is installed.
2. Open Regedit.
3. Open the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\Symantec\Backup Exec for Windows\Backup Exec\Engine\Logging

4. Create a new DWORD value named **CreateDebugLog**.
5. Right-click on the new value and click **Modify**.
6. In the **Value data** box, enter **1**.
7. Click **OK**.

Increasing the Veritas VSS provider log debug level

To increase the log debug level modify both the pre-freeze-script.bat and post-thaw-script.bat files in the `C:\Windows` folder. Add the `-log` parameter to the script, at the line where `BeVssRequestor.exe` is called. VMware determines which script is invoked.

To increase the Veritas VSS provider log debug level

1. Change the following line in the pre-freeze-script.bat:

   ```
   BeVssRequestor.exe -pre2 -logscreen !SkipExReplica! !SkipSQL!
   !VMBackupType! !ExcludeList!
   ```

   to:

   ```
   BeVssRequestor.exe -pre2 -logscreen !SkipExReplica! !SkipSQL!
   !VMBackupType! !ExcludeList! -log
   ```

2. Also change the following line in the post-thaw-script.bat:

   ```
   BeVssRequestor.exe -post2 -logscreen !SkipExReplica! !SkipSQL!
   !VMBackupType! !ExcludeList!
   ```

   to:

   ```
   BeVssRequestor.exe -post2 -logscreen !SkipExReplica! !SkipSQL!
   !VMBackupType! !ExcludeList! -log
   ```

Setting the maximum trace level for NetBackup for SQL Server

Note: For SQL Server backups, this feature is only available with legacy SQL Server backup policies.

You can set the maximum trace level in the NetBackup MS SQL Client or in the batch file. The maximum level produces large amounts of output, usually appropriate only for internal debugging.

To set the maximum trace level in the NetBackup MS SQL Client

1. Open the NetBackup MS SQL Client.
2. Select **File > Set NetBackup client properties**.
3. In the Client Trace Level group, select **Maximum**.

To set the maximum trace level in the backup or restore batch file

1. Open the NetBackup MS SQL Client.
2. Select **File > Manage script files**.
3. Select the batch file you want to change and click **Open File**.

4 Add the following line:

TRACELEVEL MAX

5 Save the file.

Troubleshooting credential validation with instance management

This topic describes the situations that may cause validation errors when you register SQL Server instances in the Applications utility.

Validation for an instance or an instance group can fail for the following reasons:

- If the host name is invalid, the following message appears:

```
Status Code: 40 Could not validate credentials. Failed to connect
to client: <client>.
```

- If the host name is correct but you cannot connect to the host because the host is down, the following message appears:

```
Status code: 46 The validation operation timed out waiting for a
response from the client
```

- If the host name is correct, but user name or password is invalid, the following message appears:

```
Status Code: 41 Validation of operating system user/password
failed for client: <client>.
```

- If the credentials do not have the “sysadmin” role, the validation fails.

```
STATUS 1939: The specified user does not have SQL Server
System Administrator privileges.
```

- If the NetBackup Client Service or the NetBackup Legacy Network Service require but do not use the same user for the logon account.

```
Invalid configuration detected. The service user for the
Netbackup Client and Netbackup Legacy Network services must be the
same user. Change the service users in the Windows Service Manager
and try again.
```

See “Configuring the NetBackup services for SQL Server backups and restores” on page 27.

- The user account does not have the required the local security privileges **Impersonate a client after authentication** and **Replace a process level token**.

  ```
  Status Code 41
  ```

  These privileges are required if you use the credentials setting **Use these specific credentials**.
  See "Configuring local security privileges for SQL Server" on page 29.

About minimizing timeout failures on large SQL Server database restores

A large SQL Server restore may fail with a Client Read Timeout error before any data has been read from the NetBackup media. This error occurs because the SQL Server may need to pre-write the database files before the restore operation begins. The time that is required for this process is a function of certain factors: the size of the database files and the speed at which your host machine can write to disk. For example, consider that your system can perform disk writes at the rate of 60 megabytes per second and you have a 2.4 terabyte database. Then it takes at least 12 hours for SQL Server to prep the disk before the actual restore can begin. In reality, the delay may be even longer than what you calculate by as much as 20% to 40%.

The timeout problem can be resolved by increasing the NetBackup Client Read Timeout setting. Use the NetBackup Administration Console on the server to change the properties of each client that contains a database you may need to restore. The default for the Client Read Timeout setting is 300 seconds (5 minutes). If you have any clients which contain large SQL Server databases, you may need to set this value much higher.

You can eliminate file initialization during SQL Server restores. See the following topic:

See "About NetBackup for SQL performance factors" on page 70.

Troubleshooting VMware backups and restores of SQL Server

Note the following when you perform a VMware backup that protects an application:

- One Application State Capture job is created per VM, regardless of which applications are selected in policy.

- The ASC job can fail if the VMware disk layout has changed since the last discovery. In this situation, you must force NetBackup to rediscover virtual machines by lowering the value of the **Reuse VM selection query results for** option. See the NetBackup for VMware Administrator's Guide.
- If the ASC job fails, the VMware snapshot or backup continues. Application-specific data cannot be restored.
  When you query the SQL Server Management Studio (SSMS), it may show that the database was backed up. In this case, though the database was skipped, the snapshot was still successful.
- Failure results in the discovery job or parent job exiting with status 1.
- ASC messages are filtered to the ASC job details.
- If you enable recovery for a particular application but that application does not exist on the VM, the ASC job returns Status 0.
- Details on the ASC job can be found in the Activity monitor job details.
- If neither the Veritas VSS provider nor the VMware VSS Provider is installed at the time of backup, the SQL Server databases are not quiesced. In this case, the recovery of a SQL Server database after it is restored may require manual steps.
- `bpfis` is executed and simulates a VSS snapshot backup. This simulation is required to gain logical information of the application.

Delays in completion of backup jobs

Sometimes you may see a NetBackup for SQL Server backup job complete the data transfer but appear to hang before the job completes. The delay may be due to one of the following:

- Network issues
- Storage transfer delays
- NetBackup server post-backup processing

To determine the cause of the delay, refer to the following article:

http://www.veritas.com/docs/TECH198864

SQL Server log truncation failure during VMware backups of SQL Server

SQL Server transaction log truncation may fail during VMware backups of SQL Server if a database name contains special characters or if the %TEMP% directory path is too long. During SQL Server log truncation, the NetBackup for SQL Server agent creates a temporary log backup. This backup specifies the current user's configured %TEMP% directory and database name as part of the destination backup device. SQL Server limits the path that can be used for backup devices to 259 characters. Under certain circumstances the SQL Server agent may generate a backup device that is longer than 259 character and cause log truncation to fail.

The following conditions cause failure:

- A configured %TEMP% directory that is longer than 259 characters.
- When the combined length of the database name and %TEMP% directory path is longer than 259 characters.

One workaround for this issue is to configure the %TEMP% directory so that the path is substantially less than 259 characters long.

SQL Server restore fails when you restore a SQL Server compressed backup image as a single stripe or with multiple stripes

This issue occurs when SQL Server is busy with the buffer of compressed data and cannot process all the data that is sent within a certain length of time. By default in Windows Server, TCP connections must close after the TCP connection state has been set to FIN_WAIT_2 for two minutes. Refer to the following Microsoft article for more information:

https://support.microsoft.com/en-us/kb/923200/

Note: If the **TCPFinWait2Delay** value does not exist, you must create it as a REG_DWORD registry value. Otherwise, Windows uses the default value of **240**.

To increase the time that TCP connections may remain in the FIN_WAIT_2 state

1. On the NetBackup media server, open `regedit.exe`.
2. Locate and select the following registry subkey:

   `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters`
3. Double-click on **TCPFinWait2Delay**.
4. Enter a value of **300**.
5. Restart the media server.
6. After the restore completes successfully, remove the registry setting or change the setting to its original value.

   When you increase the value of this setting it has an adverse effect for all TCP/IP connections. This higher value could cause port exhaustion for other applications that run on the media server.
7. Restart the media server.

Incorrect backup images are displayed for availability group clusters

You can perform backups of multiple availability group clusters that have the same short cluster name but that exist in different domains. However, it is important to use the fully qualified domain name (FQDN) of the Windows Server Failover Clustering (WSFC) cluster when you browse for backups. In the NetBackup MS SQL Client, for the **Source Client** enter the FQDN of the WSFC cluster. If you use the short cluster name, NetBackup may not display the correct list of backup images.

A restore of a SQL Server database fails with Status Code 5, or Error (-1), when the host name of the SQL Server or the SQL Server database name has trailing spaces

When the host name of a SQL Server or a SQL Server database name has one or more trailing spaces, NetBackup does not generate the restore script correctly. The trailing spaces in the SQL Server host name or the database name are truncated in the script. To successfully perform a restore, you must create and edit a restore script in the NetBackup MS SQL Client.

In the script, edit the DATABASE and the NBIMAGE lines to include the correct SQL Server host name or SQL Server database name. For example, assume that the server host name is “ACCT ”, you use the default instance, and that the database name is “DatabaseA ”. Notice the trailing spaces after the server host name and the database name.

Change the following lines:

```
DATABASE "DatabaseA"
NBIMAGE "ACCT.MSSQL7.ACCT.db.DatabaseA.~.7.001of001.20151118121736..C"
```

To:

```
DATABASE "DatabaseA   "
NBIMAGE "ACCT.MSSQL7.ACCT   .db.DatabaseA   .~.7.001of001.20151118121736..C"
```

A move operation fails with Status Code 5, or Error (-1), when the SQL Server host name, the database name, or the database logical name has trailing spaces

If the SQL Server host name, database name, or database logical name has one or more trailing spaces, a move operation fails with Status Code 5 or Error (-1). To successfully perform a move operation, you must create and edit a move script in the NetBackup MS SQL Client.

For information on a workaround for this issue, please see the following tech note on the Veritas Support website:

http://www.veritas.com/docs/000099850

Chapter 17

Disaster recovery of a SQL Server

This chapter includes the following topics:

About disaster recovery of SQL Server

SQL Server corrects itself automatically from temporary or minor problems. However, most disasters are beyond the scope of the automatic recovery feature. For example, if a database becomes severely corrupted, or there is a catastrophic failure, recovery is initiated by the system administrator.

User-initiated recovery can entail either restoring the entire server, including the SQL Server databases, from full system backups. Or recovery can include restoring only the SQL Server databases to a newly-installed or other available SQL Server.

Restoring the entire server has the added benefit of recovering other applications and data which may have resided on the server at the time of failure. Restoring be accomplished using one of the following methods:

- Manual recovery of the server. This method involves manually restoring the server from full system backups.
  See “Preparing for disaster recovery of SQL Server” on page 250.
- NetBackup Bare Metal Restore. BMR automates system recovery by restoring the operating system, system configuration, and all system files and data files. See the NetBackup Bare Metal Restore Administrator's Guide for more information.

After recovery of the server is complete, or after the new server installation is available, recovery of the SQL Server databases can begin.

Preparing for disaster recovery of SQL Server

When you develop your SQL Server disaster recovery plan you need to plan how to recover from corruption of the master database. You also need to plan for loss of your host machine. If the master database has been corrupted, then SQL Server does not start. When disaster happens you may need to rebuild the system databases. This process, however, does not recreate the schema information of your application databases. To recover your database schema use the NetBackup MS SQL Client to restore your latest backup of the master database.

Disaster recovery of SQL Server assumes that you have already put in place a strategy to recovery from other sorts of data loss. Data loss can include disk, software, and human error. To prepare for disaster recovery you need to make frequent backups of the master database. Do frequent backups after you have added or dropped databases or carried out other operations that may result in schema definitions.

Recovering SQL Server databases after disaster recovery

For the purposes of disaster recovery, you should only restore to a new installation of SQL Server. However, you can restore an existing installation of SQL Server with other active databases. The server should be running the same version of Windows on the same hardware platform. It also should be running the same version of SQL Server with the same service pack as the original server.

To recover SQL Server databases

1. If you want to restore to an existing SQL Server, choose from one of the following:
   - For a new SQL Server installation or when the master database is intact, continue with step 4.

- If the master database is corrupt, you must first rebuild the master database. Continue with step 2.

2. Refer to the following article for instructions on how to rebuild the master database. Click the “Other Versions” drop-down list to select the correct SQL Server version.

   http://msdn.microsoft.com/en-us/library/ms144259.aspx

   Look for the information that describes how to rebuild system databases for a default instance from the command prompt.

3. When the rebuild is complete, restart the SQL Server services if necessary.

4. To begin the restore of the master database, start SQL Server in single-user mode.

   The procedure to start SQL Server in single-user mode is described in the following article:

   http://msdn.microsoft.com/en-AU/library/ms188236.aspx

   Click the “Other Versions” drop-down list to select the correct SQL Server version.

5. Open the NetBackup MS SQL Client interface.

6. Locate all the media that is required to perform the restore operations.

7. Select **File > Restore SQL Server objects**.

8. Select the backup image that contains the copy of the master database you want to restore.

   Select only the master database at this time.

9. Click **Restore**.

10. Restart the SQL Server service after the restore completes.

11. Continue with the restore of the remaining SQL Server databases.

    Follow the instructions for restoring SQL databases, differentials, transaction logs, files, and filegroups.

    When all of the restore operations have completed successfully, then the recovery of the SQL Server databases is complete.

    After the recovery is complete, Veritas recommends that you perform a full database backup as soon as possible.

Appendix A

Sample batch files

This appendix includes the following topics:

About sample backup batch files for legacy SQL Server policies

Legacy SQL Server policies use batch files to initiate backup operations. These examples show you how to perform a variety of backup operation with batch files.

The following examples of batch files are available:

Script to back up a database

Certain default values define the parameters for this operation. For example, there is one backup stripe, minimum trace level, and the object type is a database (as opposed to a transaction log).

```
OPERATION BACKUP
DATABASE "BUSINESS"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
ENDOPER TRUE
```

Script to perform a striped database backup and allow multiple internal buffers per stripe

This example backs up the BUSINESS database using four data streams. Each data stream uses two buffers.

```
OPERATION BACKUP
DATABASE "BUSINESS"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
STRIPES 4
NUMBUFS 2
MAXTRANSFERSIZE 6
BLOCKSIZE 7
ENDOPER TRUE
```

Script to perform an operation and specify the user ID and password to use to SQL Server

Only specify a user ID and password if you use standard SQL Server security.

See "About SQL Server security with NetBackup legacy backup policies" on page 187.

```
OPERATION BACKUP
DATABASE "BUSINESS"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
```

```
MAXTRANSFERSIZE 6
BLOCKSIZE 7
USERID JSMITH
PASSWORD my.Pwd
ENDOPER TRUE
```

Script to perform multiple operations in sequence

In this sample batch file, five separate backups are performed sequentially. Remember that each operation is required to be completely specified.

```
OPERATION BACKUP
DATABASE "BUSINESS"
OBJECTTYPE DATABASE
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
STRIPES 5
ENDOPER TRUE

OPERATION BACKUP
DATABASE "RECREATION"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
OBJECTTYPE TRXLOG
ENDOPER TRUE

OPERATION BACKUP
DATABASE "EDUCATION"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
STRIPES 2
ENDOPER TRUE

OPERATION BACKUP
```

```
DATABASE "GOVERNANCE"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
OBJECTYPE TRXLOG
ENDOPER TRUE

OPERATION BACKUP
DATABASE "SURVIVAL"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
OBJECTYPE TRXLOG
ENDOPER TRUE
```

Script to perform a set of operations in parallel

This sample is identical to the previous sample except that the first operation contains `BATCHSIZE 3`.

See "Script to perform multiple operations in sequence" on page 254.

This setting tells NetBackup to start the first three operations in parallel. After these are completed, NetBackup then begins the next set of 3. In this case, since there are five operations, the second batch set contains two operations.

```
BATCHSIZE 3
OPERATION BACKUP
DATABASE "BUSINESS"
OBJECTTYPE DATABASE
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
STRIPES 5
ENDOPER TRUE

OPERATION BACKUP
DATABASE "RECREATION"
```

```
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
OBJECTTYPE TRXLOG
ENDOPER TRUE

OPERATION BACKUP
DATABASE "EDUCATION"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
STRIPES 2
ENDOPER TRUE

OPERATION BACKUP
DATABASE "GOVERNANCE"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
OBJECTYPE TRXLOG
ENDOPER TRUE

OPERATION BACKUP
DATABASE "SURVIVAL"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
OBJECTYPE TRXLOG
ENDOPER TRUE
```

Script to specify the maximum transfer size and block size for a backup

This sample batch file backs up database "business" with a maximum transfer size of 64 kilobytes * 2^4 (1 MB). The maximum block size is 512 bytes * 2^6 (32 KB).

```
OPERATION BACKUP
DATABASE "BUSINESS"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 4
BLOCKSIZE 6
ENDOPER TRUE
```

Script that uses environment variables to exclude instances and databases from backup

You can use SQLINSTANCE $ALL in your batch file to designate that all SQL Server instances on your host be backed up. For example, the following batch file backs up the master, model, and msdb databases. These databases are backed up on all instances of SQL Server on the host on which the batch file is run.

```
SQLINSTANCE $ALL
OPERATION BACKUP
DATABASE "master"
NBSERVER "BEARING"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
NUMBUFS 2
ENDOPER TRUE

OPERATION BACKUP
DATABASE "msdb"
NBSERVER "BEARING"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
NUMBUFS 2
ENDOPER TRUE

OPERATION BACKUP
DATABASE "model"
NBSERVER "BEARING"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
NUMBUFS 2
ENDOPER TRUE
```

To exclude SQL Server instances on your host from backup, create the Windows environmental variable NB_SQL_INSTANCE_EXCLUDE. Specify a list of instances names that you want to exclude. The list should consist of one or more names that are separated by semi-colons.

For example, use the following value to indicate that you want to exclude the default SQL Server instance and the instance named ABC-PRODUCTS from backup:

```
#DEFAULT#;ABC-PRODUCTS;
```

Note that the default SQL Server instance for the local host is designated as #default#.

You can also exclude individual databases from backup by creating a Windows environmental variable NB_SQL_DATABASE_EXCLUDE. For the value of the variable, specify a list of database names.

For example, consider the following batch file:

```
SQLINSTANCE $ALL
OPERATION BACKUP
DATABASE $ALL
NBSERVER "BEARING"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
NUMBUFS 2
ENDOPER TRUE
```

You can exclude the databases "master," "accounting," and "pubs" with the NB_SQL_DATABASE_EXCLUDE environmental variable. For the value of the variable, indicate the databases you want to exclude. Separate the database names with semi-colons.

```
MASTER;ACCOUNTING;PUBS
```

The NB_SQL_DATABASE_EXCLUDE variable is applicable only for a batch file that has DATABASE $ALL. It performs the same function as the keyword and value pair EXCLUDE <database>. If both variables are used, they augment each other to determine which databases to exclude.

About sample restore batch files

You can use batch files to initiate restore operations. The following examples of restore batch files are available:

- Script to restore a database

- Script to restore a database from multiple stripes
- Script to restore a database transaction log up to a point in time
- Script to stage a database restore from a database backup, a differential backup, and a series of transaction backups
- Script to stage a database restore from a filegroup backup, several file backups, and transaction log backups

Script to restore a database

This sample restores a database that is called pubs, based upon the following backup:

```
NBIMAGE "cadoo.MSSQL7.CADOO\SECOND.db.pubs.~.7.001of001.20140628123631..C"
```

To find out which backups you can restore, look at the `dbclient` log file created when you did the backup or by use `bplist`.

See "About using bplist to retrieve SQL Server backups" on page 227.

```
OPERATION RESTORE
OBJECTTYPE DATABASE
DATABASE "pubs"
# The following image is type: Full
NBIMAGE "cadoo.MSSQL7.CADOO\SECOND.db.pubs.~.7.001of001.20140628123631..C"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
BROWSECLIENT "CADOO"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE RECOVERED
ENDOPER TRUE
```

Script to restore a database from multiple stripes

For a striped restore, you must specify the number of stripes and the name of the first backup image name. Notice that the backup image in this example is embedded with the string `.001of004`, which indicates that it is the first of four backups.

```
OPERATION RESTORE
OBJECTTYPE DATABASE
DATABASE "Northwind"
```

```
NBIMAGE "cadoo.MSSQL7.CADOO.db.Northwind.~.0.001of004.20140216151937..C"
STRIPES 004
MAXTRANSFERSIZE 6
BLOCKSIZE 7
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
BROWSECLIENT "CADOO"
RECOVEREDSTATE RECOVERED
ENDOPER TRUE
```

Script to stage a database restore from a filegroup backup, several file backups, and transaction log backups

This example shows a script for a full database restore that you generate in the **Restore Microsoft SQL Server Objects** dialog box.

```
OPERATION RESTORE
OBJECTTYPE FILEGROUP
DATABASE "DatabaseR"
OBJECTNAME "PRIMARY"
# The following image is type: Filegroup
NBIMAGE "ca.MSSQL7.CA\SECOND.fg.DatabaseR.PRIMARY.7.001of001.20140701095634..C"
SQLHOST "CA"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CA"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE FILEGROUP
DATABASE "DatabaseR"
OBJECTNAME "DBR_FG2"
# The following image is type: Filegroup
NBIMAGE "ca.MSSQL7.CA\SECOND.fg.DatabaseR.DBR_FG2.7.001of001.20140701095425..C"
SQLHOST "CA"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CA"
```

```
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE FILE
DATABASE "DatabaseR"
OBJECTNAME "DBR_FG1_File1"
# The following image is type: File
NBIMAGE "ca.MSSQL7.CA\SECOND.fil.DatabaseR.DBR_FG1_File1.7.001of001.20140701100824..C"
SQLHOST "CA"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CA"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE FILE
DATABASE "DatabaseR"
OBJECTNAME "DBR_FG1_File2"
# The following image is type: File
NBIMAGE "ca.MSSQL7.CA\SECOND.fil.DatabaseR.DBR_FG1_File2.7.001of001.20140701100908..C"
SQLHOST "CA"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CA"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE FILE
DATABASE "DatabaseR"
OBJECTNAME "DBR_FG1_File3"
# The following image is type: File
```

```
NBIMAGE "ca.MSSQL7.CA\SECOND.fil.DatabaseR.DBR_FG1_File3.7.001of001.20140701100953..C"
SQLHOST "CA"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CA"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE TRXLOG
DATABASE "DatabaseR"
# The following image is type: transaction log
NBIMAGE "ca.MSSQL7.CA\SECOND.trx.DatabaseR.~.7.001of001.20140701100030..C"
SQLHOST "CA"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CA"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE TRXLOG
DATABASE "DatabaseR"
# The following image is type: transaction log
NBIMAGE "ca.MSSQL7.CA\SECOND.trx.DatabaseR.~.7.001of001.20140701110015..C"
SQLHOST "CA"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CA"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE RECOVERED
ENDOPER TRUE
```

Script to restore a database transaction log up to a point in time

This script is executed after the database is restored. The database is restored to the specified point in time (Feb 16, 2014 at `2:03:00` P.M.). This time precedes the date of the backup log (Feb 16, 2014 at `2:03:21` P.M.).

Note the following:

- If `STOPAT` is not specified, then the database is restored to the date of the backup log.
- You do not need to manually stage the restoration of the database backup and the associated log files. Create the script in the **Restore Microsoft SQL Server Objects** dialog box.
- Since `RECOVEREDSTATE` was not specified, the database is restored to a recovered state following successful execution of this script.

```
OPERATION RESTORE
OBJECTTYPE TRXLOG
STOPAT 20140216/14:03:00
DATABASE Northwind
NBIMAGE "cadoo.MSSQL7.CADOO.trx.Northwind.~.0.001of001.20140216140321..C"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
BROWSECLIENT "CADOO"
ENDOPER TRUE
```

Script to stage a database restore from a database backup, a differential backup, and a series of transaction backups

This example shows a script that you generate in the **Restore Microsoft SQL Server Objects** dialog box.

```
OPERATION RESTORE
OBJECTTYPE DATABASE
DATABASE "DatabaseA"
# The following image is type: Full
NBIMAGE "cadoo.MSSQL7.CADOO\SECOND.db.DatabaseA.~.7.001of001.20140701094227..C"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CADOO"
```

```
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE DATABASE
DUMPOPTION INCREMENTAL
DATABASE "DatabaseA"
# The following image is type: Full database differential
NBIMAGE "cadoo.MSSQL7.CADOO\SECOND.inc.DatabaseA.~.7.001of001.20140701103323..C"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CADOO"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE TRXLOG
DATABASE "DatabaseA"
# The following image is type: transaction log
NBIMAGE "cadoo.MSSQL7.CADOO\SECOND.trx.DatabaseA.~.7.001of001.20140701090005..C"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CADOO"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE TRXLOG
DATABASE "DatabaseA"
# The following image is type: transaction log
NBIMAGE "cadoo.MSSQL7.CADOO\SECOND.trx.DatabaseA.~.7.001of001.20140701100030..C"
SQLHOST "CADOO"
```

```
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CADOO"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE

OPERATION RESTORE
OBJECTTYPE TRXLOG
DATABASE "DatabaseA"
# The following image is type: transaction log
NBIMAGE "cadoo.MSSQL7.CADOO\SECOND.trx.DatabaseA.~.7.001of001.20140701110015..C"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CADOO"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE
```

Appendix B

Multiplexed backups

This appendix includes the following topics:

Configuring multiplexed backups of SQL Server

Multiplexing lets you interleave multiple backups to the same tape. This feature is useful if you have many simultaneous backups that use the same tape drive.

However, multiplexing can interfere with SQL Server recovery due to how SQL Server requests streams during a restore. If you enabled multiplexing for multistreamed backups, see the information on how to perform restores. To restore a multiplexed backup, you must configure the restore for one stripe.

See "Restoring multistreamed SQL Server backups" on page 98.

Configure the following to create a multiplexed backup:

- In the backup policy, select the number of **Stripes** you want to use.
  For SQL Server Intelligent policy, configure this setting on the **Microsoft SQL Server** tab. For legacy SQL Server policies, configure the **Stripes** setting when you create the backup batch file.
- In the schedules for your policy, set **Media multiplexing** to the number of backup stripes that you want to use.
  For legacy SQL Server policies, enable multiplexing in the "Application Backup" schedule.
- In the storage units that are associated with this schedule, select **Enable Multiplexing** and set **Maximum streams per drive** to the number of stripes that you want to use.

Restoring a multiplexed SQL Server backup

In most cases, Veritas does not recommend multiplexing multiple SQL Server streams from the same backup to a single tape. However, you may want to do this if you vault or export backup images. During the restore of this type of multiplexed backup, NetBackup may time out while trying to synchronize access to data blocks from the backup tape. To prevent this time out, change the stripes parameter in the recovery batch file from `STRIPES N` to `STRIPES 1`.

When you change this value it causes the restore to be performed in a single-stream. NetBackup presents the *N* backup images to SQL Server one at a time. The tape is rewound between the restore of each image.

Appendix C

Register authorized locations

This appendix includes the following topics:

Registering authorized locations used by a NetBackup database script-based policy

During a backup, NetBackup checks for scripts in the default script location and any authorized locations. The default, authorized script location for UNIX is `usr/openv/netbackup/ext/db_ext` and for Windows is `install_path\netbackup\dbext`. If the script is not in the default script location or an authorized location, the policy job fails. You can move any script into the default script location or any additional authorized location and NetBackup recognizes the scripts. You need to update the policy with the script location if it has changed. An authorized location can be a directory and NetBackup recognizes any script within that directory. An authorized location can also be a full path to a script if an entire directory does need to be authorized.

If the default script location does not work for your environment, use the following procedure to enter one or more authorized locations for your scripts. Use `nbsetconfig` to enter an authorized location where the scripts reside. You can also use `bpsetconfig`, however this command is only available on the master or the media server.

Note: One recommendation is that scripts should not be world-writable. NetBackup does not allow scripts to run from network or remote locations. All scripts must be stored and run locally. Any script that is created and saved in the NetBackup `db_ext` (UNIX) or `dbext` (Windows) location needs to be protected during a NetBackup uninstall.

For more information about registering authorized locations and scripts, review the knowledge base article:

http://www.veritas.com/docs/000126002

To add an authorized location

1. Open a command prompt on the client.
2. Use `nbsetconfig` to enter values for an authorized location. The client privileged user must run these commands.

   The following examples are for paths you may configure for the Oracle agent. Use the path that is appropriate for your agent.

   - On UNIX:

     ```
     [root@client26 bin]# ./nbsetconfig
     nbsetconfig>DB_SCRIPT_PATH = /Oracle/scripts
     nbsetconfig>DB_SCRIPT_PATH = /db/Oracle/scripts/full_backup.sh
     nbsetconfig>
     <ctrl-D>
     ```

   - On Windows:

     ```
     C:\Program Files\Veritas\NetBackup\bin>nbsetconfig
     nbsetconfig> DB_SCRIPT_PATH=c:\db_scripts
     nbsetconfig> DB_SCRIPT_PATH=e:\oracle\fullbackup\full_rman.sh
     nbsetconfig>
     <ctrl-Z>
     ```

Note: Review the NetBackup Command Reference Guide for options, such as reading from a text file and remotely setting clients from a NetBackup server using `bpsetconfig`. If you have a text file with the script location or authorized locations listed, `nbsetconfig` or `bpsetconfig` can read from that text file. An entry of `DB_SCRIPT_PATH=none` does not allow any script to execute on a client. The `none` entry is useful if an administrator wants to completely lock down a server from executing scripts.

3 (Conditional) Perform these steps on any clustered database or agent node that can perform the backup.

4 (Conditional) Update any policy if the script location was changed to the default or authorized location.