Symantec NetBackup™ for Microsoft SQL Server Administrator's Guide

for Windows

Release 7.6

Symantec NetBackup™ NetBackup for Microsoft SQL Server Administrator's Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Documentation version: 7.6

Legal Notice

This Symantec product may contain third party software for which Symantec is required to provide attribution to the third party (“Third Party Programs”). Some of the Third Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Please see the Third Party Legal Notice Appendix to this Documentation or TPIP ReadMe File accompanying this Symantec product for more information on the Third Party Programs.

Portions of this software are derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright 1991-92, RSA Data Security, Inc. Created 1991. All rights reserved.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our website at the following URL:

www.symantec.com/business/support/

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information

- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs, DVDs, or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

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. These activities include the following:

- Configuration of options for NetBackup for SQL Server operations.
- Backups and restores of databases and database components, which include transaction logs, differentials, files, and filegroups.

- Starting NetBackup for SQL Server operations from the batch files which you have created.
- Monitoring NetBackup for SQL Server operations.

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

NetBackup for SQL Server includes the following features:

- Full integration with the NetBackup master server and Media Manager.
- Stream-based backup and restore of SQL Server objects to tape or disk with SQL Server's high-speed virtual device interface.
- Snapshot-based backup and restore of SQL Server objects with NetBackup Snapshot Client methods.
- Backup and recovery of databases, differentials, files, filegroups, and transaction logs.
- Browse capability for SQL Server objects on the local nodes and remote nodes.
- Support for redirection of SQL Server restores to different locations.
- Support for multiple SQL Server instances.
- Client operation monitoring through the NetBackup Client job monitor. Server monitoring is also available through the NetBackup master.
- Performance tuning through user control of backup stripes, transfer size, and buffer usage.
- Run operations with one of the following options:
  - Immediate launch through the NetBackup MS SQL Client
  - Scheduled backup in a backup policy
  - Command-line
- Support for instances of SQL Server that are clustered with Microsoft Cluster Server or Veritas Cluster Server.
- Recovery of the Microsoft SQL Server images that were backed up with Backup Exec, through the Backup, Archive, and Restore (NetBackup Client) interface.
- Support for the following SQL Server features:
  - Page-level validation
  - Page-level recovery

  - Partial recovery
  - Database mirroring
  - Online restore
  - Partial database backup and restore
  - Optimization for read-only filegroups
  - Verify-only restore
  - Page verification during backup and restore
  - Page-level restore
- Backup and restore of the full text search catalog
- Extended object information in the SQL Server catalog
- Progress statistics during backup and restore
- Copy-only back up
- Compression and encryption of backups
- Option to retry unsuccessful backups automatically
- Ability to restore a multistream backup with use of fewer devices than it was backed up with
- 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 snapshots, and Accelerator. Only full backups are supported on these three variations of the VMware intelligent policy. There is no support for Hyper-V 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

About the NetBackup for SQL Server graphical user interface

NetBackup for SQL Server includes the following graphical user interface features:

- GUI capability to browse SQL Server databases, filegroups, and files.

- Stage a complete database recovery. This recovery can be made from the backup images that were created for databases, filegroups, files, database differentials, and transaction logs. Recovery can also be made from read-write filegroup backups and partial database backups.
- GUI assistance for page-level restore.
- Assistance to minimize backup volume thorough the use of the read-only versus read-write properties of SQL Server data.
- Restore any database objects that are backed up on one SQL Server client to another SQL Server client.
- If you do not want to perform an immediate job launch, you can create and save a backup script.
- Properties display for SQL Server databases, filegroups, and files.
- Properties display for NetBackup for SQL Server backup images.

NetBackup for SQL Server terminology

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

Table 1-1 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.)
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").

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

Term	Definition
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 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.

About VMware backups that protect SQL Server

With NetBackup, application-aware VMware backups provide a single pass backup that protects SQL Server. 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.

Help for 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`

NetBackup documentation

For a complete list of NetBackup technical documents for each supported release, see the NetBackup Release Notes, Administration, Installation, Troubleshooting, Getting Started, and Solutions Guides page at the following URL:

http://www.symantec.com/docs/DOC5332

The documents are in Adobe® Portable Document Format (PDF), viewable with the Adobe Acrobat Reader. Download the reader from http://www.adobe.com.

Symantec assumes no responsibility for the installation and use of the reader.

The NetBackup landing page on the Symantec Support website contains a list of helpful how-to and product alert topics.

Installing NetBackup for SQL Server

This chapter includes the following topics:

- Planning the installation of NetBackup for SQL
- Verifying the operating system and platform compatibility for NetBackup for SQL Server
- NetBackup server and client requirements for NetBackup for SQL Server
- Requirements for installing NetBackup for SQL Server in a NetBackup cluster
- About license keys for NetBackup for SQL Server

Planning the installation of NetBackup for SQL

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

Step	Action	Description
Step 1	Verify the operating system and platform compatibility.	See "Verifying the operating system and platform compatibility for NetBackup for SQL Server" on page 21.
Step 2	Verify the NetBackup server and client requirements for NetBackup for SQL.	See "NetBackup server and client requirements for NetBackup for SQL Server" on page 22.

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

Step	Action	Description
Step 3	Verify that the SQL Server server software is installed and operational on the NetBackup server or client.	
Step 4	If you installed SQL Server in a cluster, review the requirements for that environment.	See "Requirements for installing NetBackup for SQL Server in a NetBackup cluster" on page 22.
Step 5	Add the applicable license key(s) to the master server.	See "About license keys for NetBackup for SQL Server" on page 23.

Verifying the operating system and platform compatibility for NetBackup for SQL Server

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 Symantec Support webpage:

http://www.symantec.com/business/support/index.jsp

2 Click on the link for **NetBackup Enterprise Server**.

3 In the list of documents, click on the following document:

NetBackup Database Agent Compatibility List

4 For information on supported cluster environments for NetBackup for SQL Server, see the following document:

NetBackup Cluster Compatibility List

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

NetBackup Snapshot Client Compatibility List

6 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 for NetBackup for SQL Server

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

- The NetBackup server software is installed and operational on the NetBackup server. The NetBackup server platform can be any that NetBackup supports.
  See the NetBackup Installation Guide.
- 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 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.
  In a VMware environment, the NetBackup client software must be installed on the virtual machines that have SQL Server running.
- If the SQL client is on a different host than the master server or media server, then install the NetBackup client on that host.

Requirements for installing NetBackup for SQL Server in a NetBackup cluster

If you plan to use the database agent software on a NetBackup server configured in a NetBackup cluster, verify the following requirements:

- If NetBackup supports your cluster environment:
  See “Verifying the operating system and platform compatibility for NetBackup for SQL Server” on page 21.
- 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.
- On each node where NetBackup server resides, add the license key for the database agent.

About license keys 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 license keys.

See the NetBackup Administrator’s Guide, Volume I.

In a NetBackup cluster, add the key on each node where the NetBackup server is installed.

To use NetBackup for SQL Server and the features that are supported with the agent, you need the following keys.

Backup type or feature	License required
NetBackup for SQL Server agent	NetBackup for SQL Server
Snapshot backups	Snapshot Client
VMware backups that protect SQL Server	Enterprise Client
Replication Director	NetBackup Replication Director (this option enables Snapshot Client, OpenStorage Disk, and Replication Director)
Accelerator	Data Protection Optimization Option

Configuring NetBackup for SQL Server

This chapter includes the following topics:

Configuring the Maximum jobs per client for NetBackup for SQL Server

The following procedure shows how to set the **Maximum jobs per client** attribute.

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 99.

   The **Maximum jobs per client** specifies the maximum number of concurrent backups that are allowed per client. The default is 1.

You can 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_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.

About SQL Server security and NetBackup

NetBackup for SQL Server uses SQL 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 NetBackup for SQL Server with 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\
```

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. To have the necessary permissions, the logon account NetBackup uses for these services must have the fixed server role "sysadmin." Both services must use the same logon account.

Choose a logon account for the NetBackup services as follows:

- For SQL Server 2008 and earlier, the sysadmin role is automatically applied to the NT AUTHORITY\SYSTEM and BUILTIN\Administrators groups. You can use Local System for the logon accounts for the NetBackup services.
- For SQL Server 2012, you must first apply the sysadmin role manually to the NT AUTHORITY\SYSTEM or the BUILTIN\Administrators group. Then you can use Local System for the logon accounts for the NetBackup services.
- You can use another account as the logon account if it has the sysadmin role.
- 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 98.

Note: For an SQL Server cluster, perform the steps on each database node.

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 host and instance use standard or mixed security, perform the following steps:

- From the Windows Start menu, select **Programs > Symantec NetBackup > NetBackup Agents > 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 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

Review the following information before you create and use batch files:

- Batch files are in Unicode text.
- A batch file consists of a series of operations (backups and restores) which by default are run in sequence.

- 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.

Keywords and values used in batch files

Table 3-1 describes the keywords and values that can be used in batch files.

Table 3-1 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.

Table 3-1 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
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. Applies to all of the operations in the batch file. Must appear before the end of the first operation. Range is 1-10.
BLOCKSIZE	integer	no	0	Applicable for backup operations only. Block size is calculated as 512 bytes * $2^{\mathtt{BLOCKSIZE}}$. 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.
COPYONLY	TRUE or 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 persistent frozen image backups. See "Using copy-only snapshot backups to affect how differentials are based" on page 112.

Table 3-1 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
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	(SQL 2005 and later) 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.
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.

Table 3-1 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
GROUPSIZE	integer between 1 and 32	no	none	The number of databases that are snapped as a single SQL Server backup image. See "About NetBackup Snapshot Client for SQL Server" on page 105.
INHIBITALTBUFFER METHOD	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 * $2^{\text{MAXTRANSFERSIZE}}$. Range is 0-6.
MOVE	file group	no	none	Specifies a filegroup name. Used for restore types PARTIAL and MOVE.
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.

Table 3-1 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
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.
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	Specifies a page ID for a page restore operation.

Table 3-1 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
PARTIAL	TRUE, FALSE	no	FALSE	Specifies NetBackup perform a partial backup or restore.
PASSWORD	string	no	null	Password for logging into SQL Server. This keyword is ignored you use integrated security.
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.
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.
RESTOREPAGES	TRUE, FALSE	no	FALSE	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.

Table 3-1 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
RESTORETYPE	FULL, PARTIAL, MOVE	no	FULL	Full = Full database restore, Partial = Partial 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. 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.
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.
RETRYTYPE				See RESTARTTYPE.
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.
RETRYWAITSECONDS				See RESTARTWAITSECONDS.

Table 3-1 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 a persistent frozen image backup using the volume rollback method.
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 SQLINSTANCE is missing, then the default SQL Server instance is assumed for the SQL Host.
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.

Table 3-1 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
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 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.
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.

Table 3-1 Keywords and values used in batch files *(continued)*

Keyword	Values	Required?	Default	Description
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 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.
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 NetBackup for SQL sample batch files” on page 189.

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 58.

See "Options for NetBackup for SQL Server restores" on page 72.

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.

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 NetBackup for SQL sample batch files" on page 189.

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 host and instance" on page 57.

2 Select **File** > **Manage script files**.

3 Double-click the batch file.

4 Click **Start**.

5 To monitor the operation, select **File** > **View status**.

About configuring a backup policy for an SQL Server database

A backup policy for a database defines the backup criteria for a specific group of one or more clients.

These criteria include the following:

- Storage unit and media to use
- Policy attributes
- Backup schedules
- Clients to be backed up
- The batch files to run on the clients

To back up the database environment, define at least one MS-SQL-Server policy with the appropriate schedules. A configuration can have a single policy that includes all clients, or there can be many policies, some of which include only one client.

Most requirements for database policies are the same as for file system backups. In addition to the policy attributes for this database agent, other attributes are available that you should consider.

See the NetBackup Administrator's Guide, Volume I.

To add and configure a policy, see the following topics:

- See "Adding a new NetBackup for SQL Server policy" on page 39.
- See "About policy attributes for NetBackup for SQL Server" on page 41.
- See "About adding schedules to a NetBackup for SQL Server policy" on page 41.
- See "Adding clients to a NetBackup for SQL Server policy" on page 45.
- See "Adding batch files to the backup selections list " on page 46.

Adding a new NetBackup for SQL Server policy

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

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 48.

See "Configuring multiplexed backups of SQL Server" on page 49.

See "About configuration of SQL Server backups with a multi-interface network connection (multi-NIC)" on page 50.

To add a new NetBackup for SQL Server 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 NetBackup Administration Console, select **NetBackup Management** > **Policies**. Then select **Actions > New > 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. 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 key for the database agent.
8. Complete the entries on the **Attributes** tab.

   See "About policy attributes for NetBackup for SQL Server" on page 41.
9. Add other policy information as follows:
   - Add schedules.
     See "About adding schedules to a NetBackup for SQL Server policy" on page 41.
   - Add clients.
     See "Adding clients to a NetBackup for SQL Server policy" on page 45.
   - Add batch files to the backup selections list.
     See "Adding batch files to the backup selections list " on page 46.
10. When you have added all the schedules, clients, and backup selections you need, click **OK**.

About policy attributes for NetBackup for SQL Server

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

For more information on policy attributes, see the NetBackup Administrator's Guide, Volume I. Table 3-2 describes the policy attributes available for a NetBackup for SQL Server policy.

Table 3-2 Policy attribute descriptions for NetBackup for SQL Server policies

Attribute	Description
Policy type	Determines the types of clients that can be in the policy. In some cases the policy type determines the types of backups that NetBackup can perform on those clients. To use the SQL Server database agent, you must define at least one policy of type that is MS-SQL-Server.
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 105. See "Configuring a VMware policy to back up SQL Server using Replication Director to manage snapshot replication" on page 101.
Use Accelerator	Select this 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 100. 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.

About adding schedules to a NetBackup for SQL Server policy

Each policy has its own set of schedules. These schedules initiate automatic backups and specify when a user can initiate operations. A database backup has these types of schedules: application backup and automatic backup.

About NetBackup for SQL Server schedule properties

This topic describes the 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. Additional information about other schedule properties is available.

See the NetBackup Administrator's Guide, Volume I.

Table 3-3 Description of schedule properties

Property	Description
Type of backup	Specifies the type of backup that this schedule controls. The selection list shows only the backup types that apply to the policy you want to configure. See "NetBackup for SQL backup types" on page 42.
Schedule type	You can schedule a backup in one of the following ways: ■ **Frequency** This setting is used only for automatic schedules. It is not used for application schedules. **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 seven 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** This setting is used only for schedules. It is not used for application schedules. The **Calendar** option allows you to schedule 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.

NetBackup for SQL backup types

Table 3-4 shows the backup types you can specify.

Table 3-4 NetBackup for SQL 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 an automatic 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.
Automatic backup	An automatic backup schedule specifies the dates and times for NetBackup to automatically start backups. NetBackup runs the scripts in the order that they appear in the file list. If there is more than one client in the policy, the scripts are run on each client.

Configuring a NetBackup for SQL Server 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 SAP 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 NetBackup for SQL Server schedule properties" on page 42.

Example application backup schedule for NetBackup for SQL Server

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 3-5 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 NetBackup for SQL Server 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 **Type of backup**.

   See “NetBackup for SQL backup types” on page 42.
5. Specify the other properties for the schedule.

   See “About NetBackup for SQL Server schedule properties” on page 42.
6. Click **OK**.

Example automatic backup schedule for NetBackup for SQL Server

Table 3-6 shows example settings for an automatic backup schedule.

Table 3-6 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 NetBackup for SQL Server 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 software installed:

- SQL Server
- NetBackup client or server
- The backup or restore batch file(s)

Note: Each batch file must be present on each client.

To add clients to a NetBackup for SQL Server policy

1. To access the **Policy** dialog box, double-click the policy name in the **Policies** list in the NetBackup Administration Console.
2. In the **Policy** dialog box, click the **Clients** tab.
3. Click **New**.
4. Type the name of the client and press **Enter** (Windows) or click **Add** (Java).

   If SQL Server is installed in a cluster, specify the virtual SQL Server name as the client name.
5. (Windows) To add another client, repeat step 3 and step 4.
6. (Java) To add another client, repeat step 4.
7. If this client is the last client you want to add, click **OK**.
8. (Java) In the **Policy** dialog box, click **Close**.

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 to set up a policy for automatic scheduling. All batch files that are listed in the backup selections list are run for manual backups and for Automatic Backup schedules. The schedules are specified under 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. Make sure that the batch file resides on the client before you try to add it to the backup selections list.

To add batch files to the backup selections list from the Java interface

1. Open the policy you want to edit or create a new policy.
2. Click the **Backup Selections** tab.
3. Click **New**.
4. Specify the names of the batch files that you want NetBackup to use by doing the following:
   - In the **Script** box, type the full path name of a batch file on the client.

     For example, a path name for a batch file that backs up a database may be:

     ```
     install_path\NetBackup\DbExt\Mssql\bkup.bch
     ```

     install_path refers to the directory where you installed the NetBackup software. By default, this directory is `C:\Program Files\VERITAS\`.

     You can also specify only the file name if the batch file resides in the *install_path*\NetBackup\Dbext\Mssql\ directory.
   - Click **Add** to add the batch file to the list.
   - Add any other batch files.
5. Click **OK**.

To add batch files to the backup selections list from the Windows interface

1. Open the policy you want to edit or create a new policy.
2. Click **New**.
3. Specify the names of the batch files that you want NetBackup to use.

   Use one of the following methods:
   - Type the full path name of the batch file on the client.

     For example, a path name for a batch file that backs up a database may be:

     ```
     install_path\NetBackup\DbExt\Mssql\bkup.bch
     ```

     install_path refers to the directory where you installed the NetBackup software. By default, this directory is `C:\Program Files\VERITAS\`.

     You can also specify only the file name if the batch file resides in the `install_path\NetBackup\Dbext\Mssql\` directory.
   - Click the **Remote Folder** option.

     Navigate to and select the batch file and click **OK**.
4. Click **OK**.

Performing a manual backup for a MS-SQL-Server policy

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 Troubleshooting 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. Click the policy you want to test.
5. Select **Actions > Manual Backup**.

   The **Schedules** pane contains the name of an automatic schedule (or schedules) configured for the policy that you want to test. The **Clients** pane contains the name of the client(s) listed in the policy that you want to test.
6. Follow the directions in the **Manual Backup** dialog box.
7. 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.
8. To test compression, perform a backup with and without compression. Then compare the image sizes. The compressed image should be smaller.

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. This configuration is useful if the tape device is a bottleneck.

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.

Backup images can be written to more tapes than available drives. If this situation, in the batch file for the restore operation, indicate the number of drives that are available.

See “Restoring multistreamed SQL backups” on page 91.

To configure a multistriped backup

1. In your "Application Backup" policy, set **Media multiplexing** to **1** in the schedule you want to use for striped backups or restores.

   Then multiplexing is disabled for this schedule. All streams are made simultaneously available so restore operations are successful.

2. Ensure that the storage unit has as many drives as you want to have stripes.

3. Configure schedules for backups and restores so that enough drives are available at the time you want to perform striped backups or restores.

4. In the **Backup Microsoft SQL Server Objects** dialog box, select a **Stripes** setting greater than 1.

Configuring multiplexed backups of SQL Server

NetBackup provides a multiplexing feature that lets you interleave multiple backups to the same tape. 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.

If you use multiplexing when you perform multistreamed backups, see the information on how to perform restores.

See “Restoring multistreamed SQL backups” on page 91.

To configure multiplexed backups

1. In the schedule for your "Application Backup" policy, set **Media multiplexing** to the number of backup stripes that you plan to use.

2. In the storage units that are associated with this schedule, select **Enable Multiplexing**.

3. Set **Maximum streams per drive** to the number of stripes that you plan to use.

4. In the **Backup Microsoft SQL Server Objects** dialog box, select a **Stripes** setting equal to the number of stripes that you want to use.

About configuration of SQL Server backups with a multi-interface network connection (multi-NIC)

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 configuration requires some special considerations when you configure both the NetBackup master server and the NetBackup client that backs up SQL Server. The process to browse backup images across a required network interface is also different.

For information on how to perform restores with a multi-NIC connection, see the following topic.

See "Performing SQL restores with a multi-NIC connection" on page 88.

If you have installed SQL Server in a cluster, refer to the following topic.

See "About using SQL Server in a cluster with a multi-interface network connection (multi-NIC)" on page 147.

Configuring the master server for SQL Server with a multi-NIC

The following configuration changes must be made on the master server to allow for backups and restores over a private interface:

- The backup policies must include the private interface name in the **Clients** tab of the policy. The NetBIOS or public name of the client should not be used.
- Permissions must be added to allow for browsing of backups across the private interface.

To configure the master server for SQL Server with a multi-NIC

1. Open the NetBackup Administration Console.
2. Create a new policy or open an existing policy.
3. On the **Clients** tab for the policy, add a new client.

   Specify the Client name as the private interface name. For example, the NetBackup client computer that backs up SQL Server is SQLHOST. The private interface that is used to back up SQLHOST is SQLHOST-NB.
4. Configure one of the following:
   - Allow any NetBackup client to browse the backups the SQL Server host machine makes.
     Add the empty file `NB_INSTALL\db\altnames\No.Restrictions` on the master server.

- Allow only the SQL Server host machine, SQLHOST, to access the backups the master server manages.

  Add the empty file, `NB_INSTALL\db\altnames\`*SQLHOST* on the master server.

Configuring the SQL Server client for a multi-NIC

To configure backups of a SQL Server client in a multi-NIC environment, you must provide the following for NetBackup:

- Private interface name

  You can set client name during the installation of the NetBackup client. Or you can change it in the **NetBackup Client Properties** dialog box in the **Backup, Archive, and Restore** interface.
- Public interface name

  Set the public interface name the **NetBackup Microsoft SQL Client** interface, in the **SQL Server connection properties** dialog box.

Specifying the private name of the client for a multi-NIC

This procedure describes how to specify the private name of the client when you want to configure backups in a multi-NIC environment.

To specify the private name of the client for a multi-NIC

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 SQLHOST would be SQLHOST-NB.

Specifying the public name of the client for a multi-NIC

The following procedure describes how to specify the public name of the SQL Server client when you want to configure backups in a multi-NIC environment.

To specify the public name of the client for a multi-NIC

1. 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 SQL Server host.
4. Click **Apply** and **Close**.

About NetBackup for SQL performance factors

Your hardware environment and the settings in SQL Server, the NetBackup master server, and NetBackup for SQL Server, can all influence the backup performance. The availability of buffer space has a definite influence on how fast backups run.

The following factors can affect performance:

- NetBackup for SQL buffer space parameters
- Backup stripes
- About shared memory usage
- About alternate buffer method
- About page verification
- About instant data file initialization
- Using read-write and read-only filegroups

NetBackup for SQL buffer space parameters

Use the following parameters to increase buffer space in SQL Server:

- MAXTRANSFERSIZE
- BLOCKSIZE
- NUMBER OF BUFFERS PER STRIPE

MAXTRANSFERSIZE

The MAXTRANSFERSIZE parameter can be set for each backup or restore operation. It ranges in size from 64 kilobytes to 4 MB. The default is 4 MB. MAXTRANSFERSIZE 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.

Set the MAXTRANSFERSIZE for your NetBackup for SQL Server session by choosing **File > Set NetBackup client properties** or add this parameter to batch file.

If you use the NetBackup MS SQL Client to create a batch file, the MAXTRANSFERSIZE parameter defaults to 6. (A maximum transfer size of 4 MB.) If you remove this keyword from the batch file or manually create a batch file without this keyword, the default is 0 or a maximum transfer size of 64 KB

BLOCKSIZE

The BLOCKSIZE parameter can be set for each backup operation. For restores, NetBackup for SQL Server automatically chooses the same size that you backed up with. BLOCKSIZE ranges from 512 bytes to 64 kilobytes. The default is 64 KB. BLOCKSIZE is the incremental size that SQL Server uses for reading and writing backup images.

Set the BLOCKSIZE for your NetBackup for SQL Server session using the NetBackup client properties dialog box. Otherwise, you can use the BLOCKSIZE parameter to set this value when you create a batch file manually.

If you use the NetBackup MS SQL Client to create a batch file, the BLOCKSIZE parameter defaults to 7. (A maximum transfer size of 64 KB.) If you remove this keyword from the batch file or manually create a batch file without this keyword, the default is 0 or a maximum transfer size of 512 bytes

NUMBER OF BUFFERS PER STRIPE

The NUMBER OF BUFFERS PER STRIPE parameter affects buffer space availability. This parameter is set in the NetBackup MS SQL Client interface for each operation. NetBackup uses this parameter to decide how many buffers to allocate for reading or writing each data stream during a backup or restore 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 the NUMBER OF BUFFERS PER STRIPE is 2, which allows double buffering. You may get slightly better performance by increasing this value to a higher value.

In the Microsoft SQL Client interface, you can change this value through the NetBackup client properties dialog box. (**File > Set NetBackup client properties**). Otherwise, you can use the NUMBUFS parameter to set this value when you create a batch file manually.

Backup stripes

You can divide the backup operation into multiple concurrent streams. This configuration applies in any situation in which SQL Server dumps data faster than your tape drive is capable of writing. NetBackup does supports multiplexing many stripes to the same drive. Though you generally realize performance gains only if you use the same number of tape drives as streams.

To divide a backup operation into multiple stripes, modify the **Stripes** value. This value is in the **Backup Microsoft SQL Server objects** dialog box of the Microsoft SQL Client interface.

About shared memory usage

If NetBackup server is installed on the same host as NetBackup for SQL Server, then you see optimal performance. The optimal performance is seen if you 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.

About 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 MAXTRANSFERSIZE for the backup to the maximum allowed, which is 4 megabytes. Set this parameter by choosing File > **Set NetBackup client properties** or add this parameter to a batch file.

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 non-multiplexed.
- The backup policy does not specify either compression or encryption.
- The NetBackup buffer size equals the SQL Server blocksize.
  The default NetBackup buffer size is 64 kilobytes, 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)
  You can set the SQL Server blocksize either in the **NetBackup client properties** dialog box of the NetBackup MS SQL Client. Or you can adjust it directly using the `BLOCKSIZE` option in your batch file.
- NetBackup for SQL Server must be 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 an SQL Server backup through NetBackup for SQL Server or through `dbbackex`. Then 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.

About page verification

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

About 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 about eliminating the file initialization, 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.

Chapter 4

Performing backups of SQL Server

This chapter includes the following topics:

Overview of using NetBackup for SQL Server

You can use the NetBackup Administration Console to start SQL Server backups or use the NetBackup for SQL Server interface to initiate backups and restores.

You can perform user-directed operations for SQL Server on a Windows client in the following ways:

- Using the command line interface program, `dbbackex`.
  See "Using dbbackex to perform user-directed operations for SQL Server" on page 174.
- Using the NetBackup for SQL Server interface.
  See "Starting the NetBackup MS SQL Client for the first time" on page 57.

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. From the Windows Start menu, select **Programs > Symantec NetBackup > NetBackup Agents > 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 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 host and instance

1 From the Windows Start menu, select **Programs > Symantec NetBackup > NetBackup Agents > 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 4-1 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**.

Table 4-1 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 4-1 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 or write filegroups in a database. - **Differential on read/write filegroups** Create a differential backup of read or 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.
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 4-1 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.
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.
Stripes	Set the number of backup stripes that you want SQL Server to create for your backup. Type a number from 1 to 32. See "Configuring multistriped backups of SQL Server" on page 48. See "Configuring multistriped backups of SQL Server" on page 48.

Table 4-1 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.

Backing up SQL databases

This procedure describes how to perform a database backup.

To back up a database

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

   See “Selecting the SQL host and instance” on page 57.

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**.

   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**.

6. Select the backup options.

   See “Options for SQL Server backup operations” on page 58.

7. Click **Backup**.

8. When you are prompted to start the backup, click **Yes**.

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

Backing up SQL transaction logs

This procedure describes how to perform a transaction log backup.

To back up a transaction log

1 Set the **Recovery Model** setting to either **Full** or **Bulk-logged**.

Caution: You must ensure that the entire sequence of transaction logs generated following any database backup are maintained on the same NetBackup server. All transaction logs same facility and that none should be allowed to expire before the others. NetBackup for SQL Server requires that you follow these guidelines when you plan your backup strategy. Then you have maximum assistance when you restore your databases.

2 Change the host and instance you want to access.

See “Selecting the SQL host and instance” on page 57.

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 whose transaction logs you want to back up.

6 In the **Type of Backup** list, select **transaction log**.

7 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.

8 Select the backup options.

9 Click **Backup**.

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

Backing up SQL 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 64.

See “Backing up read-only filegroups” on page 63.

To back up a database filegroup

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

   See “Selecting the SQL host and instance” on page 57.

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

3. In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, expand the instance name.

4. Select a database whose filegroups you want to back up.

5. In the right pane, select one or more filegroups that you want to back up.

6. Select the backup options.

7. Click **Backup**.

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

Backing up read-only filegroups

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 166.

See “Viewing SQL read-only backup sets” on page 63.

Viewing SQL 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. Browse for the backup images that contain the read-only backup sets.

   See “Browsing for SQL backup images” on page 71.

2. In the **Restore Microsoft SQL Server Objects** dialog box, in the left pane, expand the instance name.

3. In the right pane, right-click the database and select **Properties**.

4. 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.

5. 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.

6. Click **OK**.

Backing up read-write filegroups

This procedure describes how to back up read-write filegroups.

Note: Backups of read-write filegroups are not supported if you have SQL Express. SQL Server Express Edition has limited support for recovering from this backup type.

To back up read-write filegroups

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

   See “Selecting the SQL host and instance” on page 57.

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**.

   Select one of the following:

   - 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. Click **Backup**.

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

Backing up SQL database files

This procedure describes how to back up database files.

To back up a database file

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

   See "Selecting the SQL host and instance" on page 57.
2. Select File > **Backup SQL Server objects**.
3. In the **Backup Microsoft SQL Server Objects** dialog box, in the left pane, expand the instance name and database.
4. In the left pane, select the filegroup that contains the files you want to back up.
5. In the right pane, select one or more files that you want to back up.
6. Select the backup options.
7. Click **Backup**.

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

Performing partial backups

This procedure describes how to perform a partial backup.

Note: Partial backups are not supported if you have SQL Express. SQL Server Express Edition has limited support for recovering from this backup type.

To perform a partial backup

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

   See "Selecting the SQL host and instance" on page 57.
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 a database that you want to back up.
5. Select the **Type of Backup**.

   Select one of the following:

   - To perform a full backup, select **Create a template for partial backup**.
   - To perform a differential backup of the read-write filegroups, select **Create a template for partial differential backup**.

6 Select the backup options.

7 Click **Backup**.

8 In the **Save Script As** dialog box, specify a file name and click **OK**.

9 When you are prompted to open the template, click **Yes**.

10 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
```

11 When you are finished modifying the template, save it.

12 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 installations

1 Select the host and instance you want to access.

See “Selecting the SQL host and instance” on page 57.

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

3 Select the options for the operation.

See "Options for SQL Server backup operations" on page 58.

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 and add the batch file to the **Backup Selections** list in the policy.

About file checkpointing with NetBackup for SQL

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:

MAXRESTARTS *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 "'backup 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 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” on page 67.

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 5

Performing restores of SQL Server

This chapter includes the following topics:

- About full-text catalog directory structure
- Restoring multistreamed SQL backups

Browsing for SQL 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 object.

To browse for backup images

1 Change the host and instance you want to access.

See "Selecting the SQL host and instance" on page 57.

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.

Also indicate the **Source Client**, if applicable. See the following descriptions.

SQL Host From this list, select the SQL Server host that you want to generate a backup history for.

Source Client Specifies the NetBackup client name for the selected client. This name is the one used in the policy to identify the client.

If you use a specific network interface for backup, then the network interface name should be entered in the Source Client box. The network interface name is defined in the Host Properties for the server (open the properties for the server and click **Universal Settings**).

4 Select the date range to search.

5 Click **OK**.

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

See “Restoring a SQL database backup” on page 75.

See “Staging a full SQL database recovery” on page 76.

See “Restoring SQL filegroup backups” on page 77.

See “Recovering a SQL database from read-write backups” on page 78.

See “Restoring read-only filegroups” on page 78.

See “Restoring SQL database files” on page 79.

See “Restoring a SQL transaction log image without staging a full recovery” on page 79.

See “Performing a SQL database move” on page 80.

See “About performing a page-level restore” on page 84.

Options for NetBackup for SQL Server restores

Table 5-1 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 5-1 Options for restore operations

Option	Description
Use replace option	Restore with the SQL Server replace option.
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 5-1 Options for restore operations *(continued)*

Option	Description
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 5-1 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.
MM, YYYY, 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

Table 5-1 Options for restore operations *(continued)*

Option	Description
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
Launch immediately Restore script	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 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 backup images" on page 71.

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 72.

6 Click **Restore**.

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

Staging a full SQL 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 backup images” on page 71.

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 filegroup backups

You can periodically back up only read-write filegroups as a part of your backup strategy. If periodically is the method you use, see the instructions for that type of restore.

See "Recovering a SQL database from read-write backups" on page 78.

Information is also available for how to restore read-only filegroups.

See "Restoring read-only filegroups" on page 78.

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 database recovery" on page 76.

To restore a filegroup backup

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

   See "Browsing for SQL backup images" on page 71.

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 72.

5. Click **Restore**.

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

Recovering a SQL database from read-write 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 backup images” on page 71.

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 read-only filegroups” on page 78.

Restoring 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 backup images” on page 71.

   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 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 backup images" on page 71.

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 72.

7 Click **Restore**.

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

Restoring a SQL 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 backup images" on page 71.

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 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.

To perform a database move

1 Browse for the backup images you want to restore.

See "Browsing for SQL backup images" on page 71.

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 or partial database restore 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 72.

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**.

Restoring partial databases (SQL Server 2000)

This topic shows you how to perform a partial database restore on a SQL Server 2000 database. A partial database restore lets you select individual filegroup components of a database and restore them to a new database having a different name.

Note: When you prepare your database design, map logically related database components such as tables, indexes, and keys to the same filegroup. This design enables you to use the partial database restore procedure to recover these logically related components as a single unit.

To perform a partial database restore

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

   See "Browsing for SQL backup images" on page 71.

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.

5. From the **Scripting** list, select **Create a partial database restore template**.

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

6. Click **Restore**.

7. In the **Save Script As** dialog box, type the name of a file to which you want NetBackup to write the template and click **Save**.

8. Click **Yes** to open the template in Notepad.

9. Replace the database name in the template with the name of the target database of the restore.

   For example, replace:

```
#DATABASE "DatabaseA"
```

   with:

```
DATABASE "DatabaseB"
```

10. To restore a file, indicate a new file path and uncomment the `MOVE` and `TO` lines.

    For example, replace:

```
#MOVE  "DBA_FG1_File1"
#TO  "C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DBA_FG1_File1.ndf"
```

    with:

```
MOVE  "DBA_FG1_File1"
TO  "C:\Microsoft SQL Server\MSSQL.3\MSSQL\DATA\DBB_FG1_File1.ndf"
```

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

12 When you finish modifying the template, save it.

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

About performing a page-level restore

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, 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 200x 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 200x Management Studio.

2 Browse for the backup images you want to restore.

See "Browsing for SQL backup images" on page 71.

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.

Redirecting a SQL restore to a different host

Multiple NetBackup database clients can be configured to use a single NetBackup master server for backing up SQL Server databases. With this configuration, you can back up SQL Server database objects from one client and restore them to another. In the following procedure, `HostA` is the client that was originally backed up and `HostB` is the other client to which the restore is redirected.

To redirect a restore to a different host

1 Establish permission settings on the master server.

Do one of the following:

- Create a *`install_path`*`\NetBackup\db\altnames\No.Restrictions` file to allow unrestricted redirected restore privileges.
- Create a *`install_path`*`\NetBackup\db\altnames\HostB` file to allow `HostB` to restore `HostA`'s data. The client named `HostB` can then access `HostA`'s data on the master as well as any other client's data that was backed up on the master.

2 Select File > **Set SQL Server connection properties**.

3 Log onto the target instance on `HostB`.

4 Select File > **Set NetBackup client properties** and select the current master server to be a master server that is common to both `HostA` and `HostB`.

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

6 From the Backup History Options dialog box, select `HostA` as the SQL host from which the target objects were backed up.

When the restore window appears it contains a history of objects that were backed up from `HostA`. Select those objects and restore them to `HostB`.

Redirecting a SQL database to a different location on a different host

The database move redirects the restore of a SQL Server database to a different location. 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.

Note: The destination host and instance of a move or restore operation is the one that you log into. For move or restore operations the source (or browse) host and the instance is designated in the **Backup History Options** dialog box. This dialog box opens automatically when you select **File > Restore SQL Server objects**.

To redirect a database to another location on a different host

1. 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 server" on page 88.

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

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

5. Click **Apply** and then **Close**.
6. Select **File > Set NetBackup client properties**.
7. 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.
8. Click **OK**.
9. Select **File > Restore SQL Server objects**.
10. In the **Backup History Options** dialog box, in the **SQL Host** list, select the host that has the database you want to restore.
11. Indicate the **Source Client**, if applicable.

    This name is used in the policy to identify the client.

    If you use a specific network interface for backup, enter the network interface name in the Source Client box. The network interface name is defined in the Host Properties for the server (open the properties for the server and click **Universal Settings**).

12. Click **OK**.
13. Browse for the database that you want to move.
14. From the **Scripting** list, select **Create a move template**.

15 Click **Restore**.

16 In the **Save As** dialog box, enter a file name and click **Save**.

17 Click **Yes** to open the template.

18 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.

19 After you edit the template, select **File > Manage script files** to launch it.

See "Redirecting a SQL restore to a different host" on page 86.

See "About selecting a server" on page 88.

About selecting a 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 database move" on page 80.

Performing SQL restores with a multi-NIC connection

To perform restores with a multi-NIC connection, you need to specify the public name of the SQL Server host in the **SQL Server connection properties** dialog box. You also need to provide both the SQL Server host name and the private interface name in the **Backup History Options** dialog box.

To perform restores with a multi-NIC connection

1 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 SQL Server host (SQLHOST).

4 Click **OK**.

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

6 In the **SQL Host** box, specify the primary (or NetBIOS) name.

7 In the **Source Client** box, specify the private interface name.

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.

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 host and instance" on page 57.

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

3 Select the options for the operation.

See "Options for NetBackup for SQL Server restores" on page 72.

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 full-text catalog directory structure

If your databases contain full-text search catalogs, then they are backed up with the databases and are restored as a part of the recovery process. You can also find additional information about full-text catalogs to help you understand how the backup and restore processes work.

A full-text catalog consists of a root directory, subdirectories, and dozens of files. SQL Server names the root full-text directory that is based on the directory name the end user gives. Typically, the directory name is the same as the index name, but SQL Server may alter if there are name conflicts. For example, consider if there is a full-text catalog named ‘cats’ for database ONE. Then a user creates a second database named ‘cats’ for database TWO. SQL Server creates a root full-text directory named ‘cats0000’ for the second full-text catalog.

SQL Server also creates a logical file to contain full-text catalogs. The logical file is named as `sysft_`*`full-text catalog name`*. For example, in the previous example, the "cats" catalog would be contained by `sysft_cats`. This name has import for the database move operation.

SQL Server places the full-text catalog directory into the following path by default:

```
SQL-instance-install-path\FTData\full-text-catalog-root-directory.
```

For instance, in the previous example, the cats directory for database TWO is placed into a path such as:

```
C:\Program Files\Microsoft SQL Server\MSSQL.1\FTData\cats0000\
```

However, when you perform a database move command, you can establish the path at the location of your choosing.

One additional characteristic of the full-text catalog is its association with a filegroup. By default, the association is made with the primary filegroup. However, you can also select a different filegroup for which to make the association.

Backing up and restoring the databases that contain full-text catalogs

When you back up a database that contains a full-text catalog, NetBackup writes the catalog name and directory path to NetBackup catalog. NetBackup for SQL Server displays the full-text catalog information in the backup history properties windows when you select the database backup image. If you choose to move the database, then the saved full-text catalog information allows NetBackup for SQL Server to generate specifications for the move template. In the move template, you can specify a path for the catalog directory.

Snapshot backups also include all of the files in any full-text catalog that is associated with the object that is backed up. This affects backups database, filegroup, and file backups as follows:

- Database backup: The full-text catalog files are backed up with the full database snapshot backup.
- Filegroup backup: The full-text catalog files are backed up for any catalog that is associated with the filegroup.

- File backup: NetBackup for SQL Server backs up the full-text catalog files along with a snapshot file backup

  The backup is performed based on the following conditions:

  - The file belongs to a filegroup that is associated with a full-text catalog
  - The file has the lowest sort-order of any file that belongs to the filegroup.

Backups and restores based on snapshots are affected based on the extra catalog files that are included in the snapshot.

Moving a database that contains full-text catalogs

NetBackup for Microsoft SQL Server extends the move template when a database contains a full-text catalog by including the following text:

```
MOVE  "sysft_HisCatalog"
#TO  "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\FTData\HisCatalog"
```

Modify the target directory path in the template. The name `sysft_HisCatalog` is the logical name of the file that contains the full-text directory named `HisCatalog`. NetBackup creates this name by prepending the catalog name with `"sysft_"`.

Restoring multistreamed SQL backups

When you use the NetBackup SQL Client, backups using multiple stripes are restored automatically. 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.~.7.001of003.20060908200234..C
juneberry.MSSQL7.COLE.db.pubs.~.7.002of003.20060908200234..C
juneberry.MSSQL7.COLE.db.pubs.~.7.003of003.20060908200234..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 177.

About advanced 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: Since Persistent Frozen Image (PFI) policies do not stream the component files to tape, PFI backups ignore the multistream directive.

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.~.7.001of003.20061012131132..C
```

and are differentiated by NetBackup by their backup IDs.

Restoring a multiplexed SQL Server backup

In most cases, Symantec 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.

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 6

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.
- 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 Symantec VSS provider for vSphere

Symantec recommends that you use the Symantec VSS provider. The Symantec provider truncates logs on SQL Server virtual machines by means of full VSS backups.

When the Symantec VSS provider is installed and NetBackup starts a virtual machine snapshot, VMware Tools calls the Symantec 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 Symantec VSS provider must be installed separately.

See "Installing the Symantec VSS provider for vSphere " on page 97.

Limitations of using a VMware policy to protect SQL Server

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

- 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.
- A SQL Server database name cannot contain any of the following characters: ? * \ "
- 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.
  - Select the **Exclude data disks** option.

- 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) and GPT (GUID partition table) disks.
    If NetBackup detects any database objects on a VHD or a GPT disk, the ASC job fails and no SQL Server content is cataloged. This exclusion includes any objects that do not exist on the VHD or the GPT disk.
  - The ASC job detects an excluded Windows boot 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:).

About configuring NetBackup for VMware backups that protect SQL Server

Table 6-1 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. On each ESX server that hosts the database, install the NetBackup for SQL license and the Enterprise Client license. Install the NetBackup client software on the virtual machines that have SQL Server running.

Table 6-1 Steps to configure VMware backups that protect SQL Server *(continued)*

Step	Action	Description
Step 2	Install the Symantec VSS Provider.	The Symantec 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 Symantec VSS provider for vSphere ” on page 97.
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 the following requirements: ■ The account has the fixed server role “sysadmin.” ■ Both services must use the same logon account. ■ If you want to use Replication Director, you must configure the services with a logon account that has access to the NetApp disk array. See “Configuring the NetBackup services for VMware backups that protect SQL Server” on page 98.
Step 4	(SQL 2012 on Windows 2012 or Windows 2008) 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.
Step 6	Configure a VMware policy.	See “Configuring a VMware backup policy to protect SQL Server” on page 100. See the NetBackup for VMware Administrator's Guide.

Installing the Symantec VSS provider for vSphere

To use the Symantec 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 Symantec VSS provider

1. Browse to the following location:

   `install_path\Veritas\NetBackup\bin\goodies\`

2. Double-click on the **Symantec 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 Symantec VSS provider

1. In the Control Panel, open **Add or Remove Programs** or **Programs and Features**.
2. Double-click on **Symantec 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 on Windows 2012 or Windows 2008) 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: For an SQL Server cluster, perform the steps on each database node.

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 necessary requirements:
2. If the SQL Server host and instance use standard or mixed security, perform the following steps:
   - From the Windows Start menu, select **Programs > Symantec NetBackup > NetBackup Agents > 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 NetBackup for VMware Administrator's Guide.
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 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 Enable or disable the **Enable block-level incremental backup** option as follows:

- Enable this option if you selected **Use Accelerator**.
- Disable this option if you do not use NetBackup Accelerator.

17 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.

18 (Optional) Select **Truncate logs**.

For SQL, this option truncates the transaction logs when the VMware snapshot of the virtual machine is complete. If you do not check **Truncate logs**, the logs are not truncated.

Note: If you want to truncate logs, you must first perform a full VMware backup without log truncation. Without this initial full backup, the ASC job fails. When the backup is complete, then perform the full VMware backup with log truncation enabled. To create a full backup, you must use the Symantec VSS provider.

19 Click **OK**.

Configuring a VMware policy to back up 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. For log truncation, the Symantec VSS Provider must be installed.

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 (Optional) Select **Truncate logs**.

For SQL, this option truncates the transaction logs when the VMware snapshot of the virtual machine is complete. If you do not check **Truncate logs**, the logs are not truncated.

Note: If you want to truncate logs, you must first perform a full VMware backup without log truncation. Without this initial full backup, the ASC job fails. When the backup is complete, then perform the full VMware backup with log truncation enabled. To create a full backup, you must use the Symantec VSS provider.

17 Click **OK**.

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:

- If you want to truncate logs, you must first perform a full VMware backup without log truncation. Without this initial full backup, the ASC job fails. When the backup is complete, then perform the full VMware backup with log truncation enabled.
- To create a full backup, you must install the Symantec VSS provider. The VMware VSS provider creates copy-only backups, which cannot be used as a basis to truncate logs.
- For SQL 2012 on Windows 2012 or Windows 2008, 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.

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 backup images” on page 71.

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 7

Using NetBackup for SQL Server with Snapshot Client

This chapter includes the following topics:

About NetBackup Snapshot Client for SQL Server

The NetBackup Snapshot Client feature can be used with the NetBackup SQL Agent to employ snapshot technology to back up and restore SQL Server objects. To use NetBackup Snapshot Client with NetBackup for SQL Server, you must have a license key for both Snapshot Client and Microsoft SQL Server.

Snapshot Client, when used with NetBackup for SQL Server, backs up and restores SQL Server objects with snapshots of the component files. Then it backs up the snapshot version to tape or commits it to another form of persistent storage. The snapshot technology uses SQL Server VDI (virtual device interface) quiescence to affect a momentary freeze on database activity. Then NetBackup takes a snapshot of the identified files.

The term database freeze is interchangeable with database quiescence. It refers to the momentary state in which all activity in the database has been halted while a snapshot (or frozen image) copy is created. The state-change in which the freeze is lifted is referred to as database thaw.

The role of NetBackup for management of the backup images that were taken from snapshots is common for both Snapshot Client and standard database backups. NetBackup provides an identical view for cataloging and browsing images and for accessing them in restore operations. It is transparent whether standard database backups or the Snapshot Client created the images. Operational experience is similar whether the user backs up and restores standard SQL Server database images or any images that were taken from snapshots. However, there are fundamental differences in the technologies.

Snapshot Client backups, in contrast to standard backups, are file-based. NetBackup determines the file list that constitutes the SQL Server object and backs it up asynchronously with respect to SQL Server. On the other hand, standard backups are stream-based, which means that SQL Server provides data to NetBackup buffer-by-buffer that constitutes a backup stream. The key role of SQL Server in file-based backups is to provide the mechanism to freeze database activity. NetBackup can then invoke a so-called snapshot provider that creates volume snapshots of the files.

The snapshot method is the software component or hardware component that creates the snapshot image of the files. The method may be a Symantec product such as Volume Manager or an operating system component such as VSS (Microsoft Virtual Shadow Service). Ia may also be a third-party product. Snapshot methods use a variety of technologies such as copy-on-write, split-mirror, or RAID. They also provide several competing methodologies. These technologies may have different consequences for recovery speed: effect of the backup on the client processor, and the cost of specialized disks or processing units.

About Snapshot Client features

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

Snapshot backup	A snapshot is a disk image of the client's data that is made almost instantaneously. NetBackup backs up the data from the snapshot image, not directly from the client’s primary data.
Instant Recovery (Persistent Frozen Image or PFI)	An Instant Recovery backup is similar to a snapshot backup except that the snapshot is not transferred to tape. The restore may be made directly from snapshot copy.

Persistent Frozen Image with Backup to Tape	This feature is similar to the Instant recovery feature except that a backup copy is made to a tape storage unit. If the image is needed for a restore, then a disk copy is used unless it has been swapped out due to storage contention. In this case, the restore falls back to the tape copy of the backup.
Off-host alternate client backup	An off-host alternate client backup shifts the burden of backup processing to an alternate client. This action reduces the effect on the client's resources ordinarily caused by a local backup. The backup agent reads the data from the SharedDisk and writes it to storage.

Although all of these features are provided through Snapshot Client support for SQL Server, not all 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 supported platforms matrix on the Symantec Support website.

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

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.

Another consideration for choosing between file- and stream-based backups concerns how SQL Server zeroes the target disk area before a stream-based restore. In some cases, the time almost equals the total disk-copy time for restore. For Snapshot Client restores, however, disk zeroing is not done so the total recovery time can be substantially less.

Review the other considerations for SQL Server disk initialization.

See “About NetBackup for SQL performance factors” on page 52.

Performing SQL Server snapshot backups

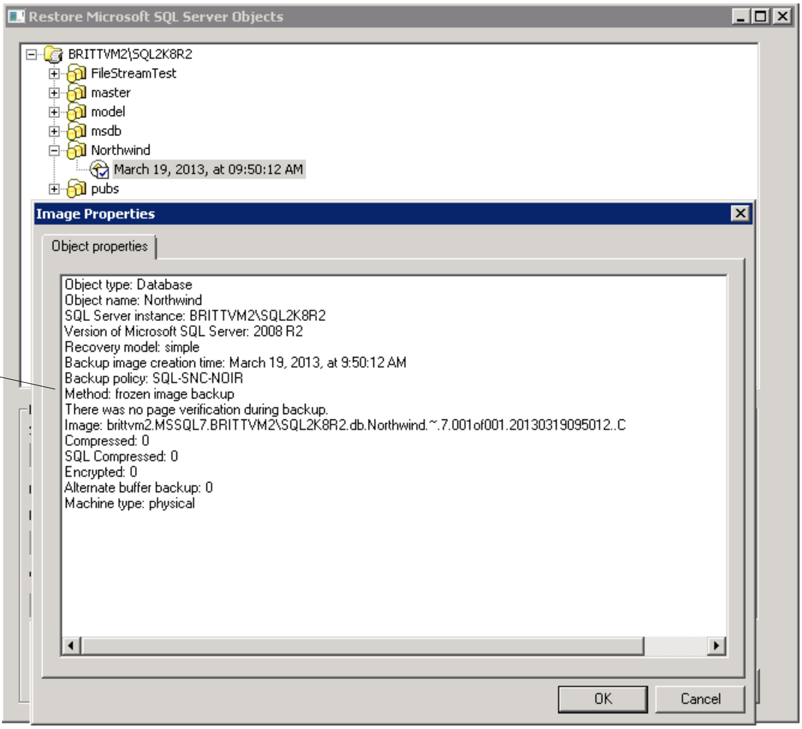
No special interfacing considerations exist when you perform Snapshot Client backups of SQL Server. A snapshot backup is performed if the backup object is: a database, a filegroup, or a file and a policy is selected and configured for Snapshot

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.

Performing 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 7-1 Backup method that appears in the backup image properties



```
Object type: Database
Object name: Northwind
SQL Server instance: BRITTVM2\SQL2K8R2
Version of Microsoft SQL Server: 2008 R2
Recovery model: simple
Backup image creation time: March 19, 2013, at 9:50:12 AM
Backup policy: SQL-SNC-NOIR
Method: frozen image backup
There was no page verification during backup.
Image: brittvm2.MSSQL7.BRITTVM2\SQL2K8R2.db.Northwind.~.7.001of001.20130319095012..C
Compressed: 0
SQL Compressed: 0
Encrypted: 0
Alternate buffer backup: 0
Machine type: physical
```

Indicates backup was created with a frozen image (snapshot) method.

About SQL Server agent grouped backups

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.

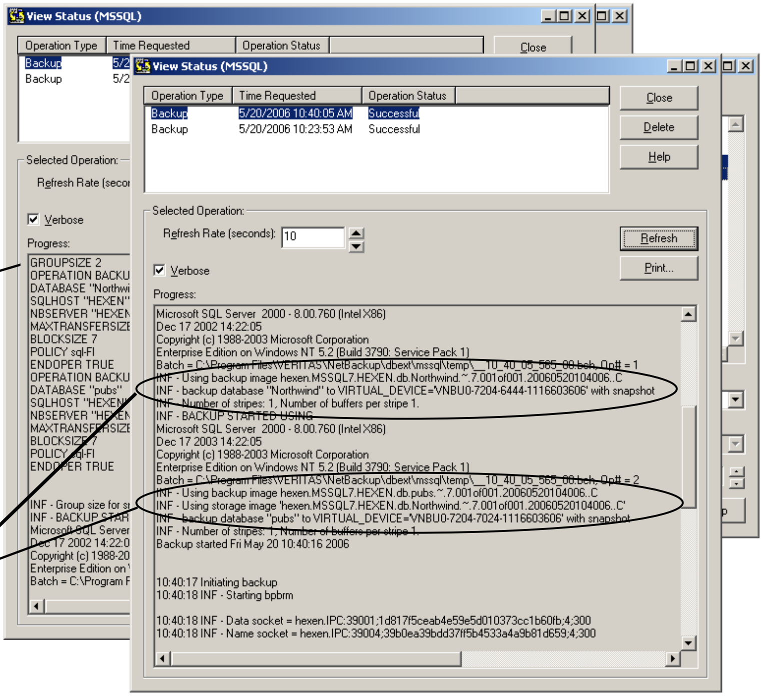
Viewing the progress of a grouped backup

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

See Figure 7-2.

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 7-2 Progress report for a grouped backup operation



Operation Type	Time Requested	Operation Status
Backup	5/20/2006 10:40:05 AM	Successful
Backup	5/20/2006 10:23:53 AM	Successful

```
GROUPSIZE 2
OPERATION BACKU
DATABASE "Northwi
SQLHOST "HEXEN"
NBSERVER "HEXEN
MAXTRANSFERSIZE
BLOCKSIZE 7
POLICY sql-FI
ENDOPER TRUE
OPERATION BACKU
DATABASE "pubs"
SQLHOST "HEXEN
NBSERVER "HEXEN
MAXTRANSFERSIZE
BLOCKSIZE 7
POLICY sql-FI
ENDOPER TRUE


INF - Group size for s
INF - BACKUP STAR
Microsoft SQL Server
Dec 17 2002 14:22:0
Copyright (c) 1988-20
Enterprise Edition on
Batch = C:\Program F
```

```
Microsoft SQL Server  2000 - 8.00.760 (Intel X86)
Dec 17 2002 14:22:05
Copyright (c) 1988-2003 Microsoft Corporation
Enterprise Edition on Windows NT 5.2 (Build 3790: Service Pack 1)
Batch = C:\Program Files\VERITAS\NetBackup\dbext\mssql\temp\__10_40_05_565_00.bch, Op# = 1
INF - Using backup image hexen.MSSQL7.HEXEN.db.Northwind.~.7.001of001.20060520104006..C
INF - backup database "Northwind" to VIRTUAL_DEVICE='VNBU0-7204-6444-1116603606' with snapshot
INF - Number of stripes: 1, Number of buffers per stripe 1.
INF - BACKUP STARTED USING
Microsoft SQL Server  2000 - 8.00.760 (Intel X86)
Dec 17 2003 14:22:05
Copyright (c) 1988-2003 Microsoft Corporation
Enterprise Edition on Windows NT 5.2 (Build 3790: Service Pack 1)
Batch = C:\Program Files\VERITAS\NetBackup\dbext\mssql\temp\__10_40_05_565_00.bch, Op# = 2
INF - Using backup image hexen.MSSQL7.HEXEN.db.pubs.~.7.001of001.20060520104006..C
INF - Using storage image 'hexen.MSSQL7.HEXEN.db.Northwind.~.7.001of001.20060520104006..C'
INF - backup database "pubs" to VIRTUAL_DEVICE='VNBU0-7204-7024-1116603606' with snapshot
INF - Number of stripes: 1, Number of buffers per stripe 1.
Backup started Fri May 20 10:40:16 2006


10:40:17 Initiating backup
10:40:18 INF - Starting bpbrm

10:40:18 INF - Data socket = hexen.IPC:39001;1d817f5ceab4e59e5d010373cc1b60fb;4;300
10:40:18 INF - Name socket = hexen.IPC:39004;39b0ea39bdd37ff5b4533a4a9b81d659;4;300
```

Indicates that this set of databases is a candidate to be backed up as a group.

Backup image name and storage image name is shown in a grouped backup.

Restoring a database backed up in a group

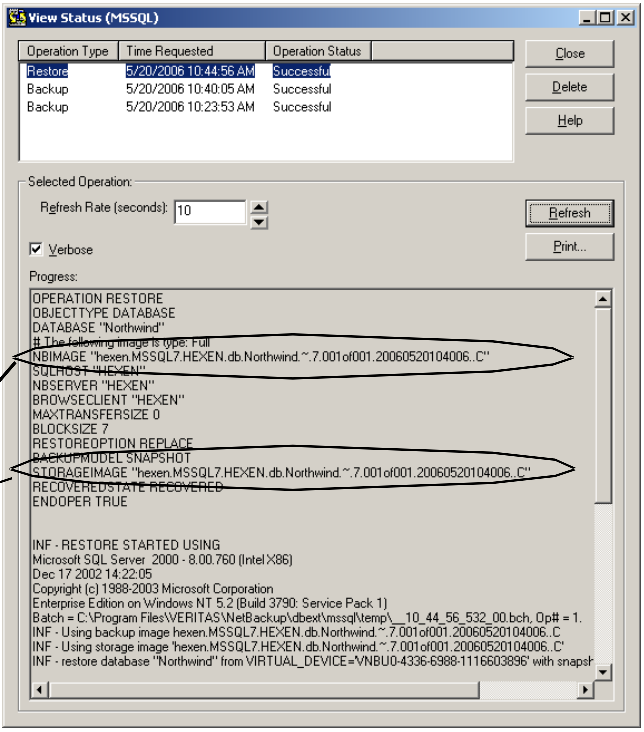
A database that is backed up in a group can be restored like any other database.

See "Restoring a SQL database backup" on page 75.

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

See Figure 7-3 on page 112.

Figure 7-3 Batch file shown in the progress report for the restore operation



Operation Type	Time Requested	Operation Status
Restore	5/20/2006 10:44:56 AM	Successful
Backup	5/20/2006 10:40:05 AM	Successful
Backup	5/20/2006 10:23:53 AM	Successful

Selected Operation:

Refresh Rate (seconds): 10

☑ Verbose

Progress:

```
OPERATION RESTORE
OBJECTTYPE DATABASE
DATABASE "Northwind"
# The following image is type: Full
NBIMAGE "hexen.MSSQL7.HEXEN.db.Northwind.~.7.001of001.20060520104006..C"
SQLHOST "HEXEN"
NBSERVER "HEXEN"
BROWSECLIENT "HEXEN"
MAXTRANSFERSIZE 0
BLOCKSIZE 7
RESTOREOPTION REPLACE
BACKUPMODEL SNAPSHOT
STORAGEIMAGE "hexen.MSSQL7.HEXEN.db.Northwind.~.7.001of001.20060520104006..C"
RECOVEREDSTATE RECOVERED
ENDOPER TRUE


INF - RESTORE STARTED USING
Microsoft SQL Server  2000 - 8.00.760 (Intel X86)
Dec 17 2002 14:22:05
Copyright (c) 1988-2003 Microsoft Corporation
Enterprise Edition on Windows NT 5.2 (Build 3790: Service Pack 1)
Batch = C:\Program Files\VERITAS\NetBackup\dbext\mssql\temp\__10_44_56_532_00.bch, Op# = 1.
INF - Using backup image hexen.MSSQL7.HEXEN.db.Northwind.~.7.001of001.20060520104006..C
INF - Using storage image 'hexen.MSSQL7.HEXEN.db.Northwind.~.7.001of001.20060520104006..C'
INF - restore database "Northwind" from VIRTUAL_DEVICE='VNBU0-4336-6988-1116603896' with snapsh
```

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

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 persistent frozen image backups, it may not be feasible to retain the daily full backup after the next full backup has been created. If a point in time restore is required before the latest backup, the differentials can effectively be based on the backups that no longer exist. The only alternative would be to recover based upon the last full backup that had been retained. You would 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
- A persistent frozen image backup is created several times per day and expires when the next one is created. This backup is created as copy-only.

Table 7-1 shows an excerpt from this schedule.

Table 7-1 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
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

Table 7-1 Sample backup schedule using copy-only backups *(continued)*

Time	A full backup saved to secondary storage	Differential backup	PFI Copy-Only	Transaction log backup
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 persistent frozen image 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

Any backup can be created as copy-only. A persistent frozen image snapshot is automatically created as copy-only. You can also create a snapshot that is NOT copy-only.

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 a PFI snapshot that is not copy-only

If a persistent frozen image snapshot is selected for backup, NetBackup automatically creates the backup image as copy-only. You can choose NOT to have the backup created as copy-only.

To create a PFI snapshot 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.

Configuration requirements for SQL 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 Symantec 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 the license key for this option has been registered. See the NetBackup Snapshot Client Administrator's Guide for details.
- Only one snapshot method can be configured per policy. If you want to select one snapshot method for clients, one group of clients, and a different method for another group, then you would need to create two policies for each group of clients. 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. Information on other policy attributes, how to schedules, how to add clients, and how to add backup selections is described in other topics.

See "About configuring a backup policy for an SQL Server database" on page 39.

To configure a snapshot policy for SQL Server

1. Create a backup script (`.bch` file) using the NetBackup for SQL Server 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 unit/lifecycle policy**.

   If database differentials, filegroup differentials, or transaction logs are included in the **Backup Selections** list of a Snapshot Client policy, 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 Click the **Schedules** tab.

9 Configure an Application and an Automatic schedule.

10 (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.

11 Use the **Clients** tab to specify clients to be backed up by this policy.

12 Use the backup selections tab to enter the batch files.

13 Click **OK** to close the dialog box.

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 snapshot backups for a MS-SQL-Server policy. Information on other policy attributes, how to schedules, how to add clients, and how to add backup selections is described in other topics.

See “About configuring a backup policy for an SQL Server database” on page 39.

To configure a policy for Instant Recovery

1 Create a backup script (`.bch` file) using the NetBackup for SQL Server 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 unit/lifecycle policy**.

If you select an Instant Recovery option on the **Schedules** tab (see step 9), the storage unit is not used.

NetBackup creates only a disk snapshot.

If database differentials, filegroup differentials, or transaction logs are included in the **Backup Selections** list of a Snapshot Client 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 9).

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

10 Follow the instructions to configure an Application and Automatic schedule.

11 (Optional) To create a disk image only, open the Application schedule 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 Use the **Clients** tab to specify clients to be backed up by this policy.

13 Use the backup selections tab to add the batch files.

14 Click **OK** to close the dialog box.

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 121.
- 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 are developing 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 need to use the SQL Server rebuild master (`rebuildm.exe`) utility to start the SQL Server service. This utility, however, does not recreate the schema information of your application databases. To recover your database schema use the NetBackup SQL Server agent 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, minimally. 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 Service Pack level: 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, you must first rebuild the master database.

- Run the Rebuild Master utility (`\`*`SQL Server installation directory`*`\bin\Rebuildm.exe`).

Note: See the Microsoft SQL Server official documentation for information on how to use `rebuildm.exe`.

- When the rebuild is complete, restart the SQL Server services if necessary.

2 If you installed SQL Server in a cluster, you need to start SQL Server in single-user mode before restoring the databases.

  - In the Windows Control Panel, open **Services**.
  - Select the MSSQLServer service.
  - Type the following in the Startup Parameters box:
  - **/m**
  - Click **Start**.

3 Open the NetBackup for SQL Client interface.

4 Locate all the media that is required to perform the restore operations.

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

6 Select the backup image that contains the copy of the master database you want to restore.

  Select only the master database at this time.

7 Click **Restore**.

8 If SQL Server is configured in a cluster, restart the SQL Server service after the restore has completed.

9 Continue restoring 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, Symantec recommends that you perform a full database backup as soon as possible.

SQL backups and restores in an SAP environment

This chapter includes the following topics:

About backups and restores in an SAP environment

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. From the Windows Start menu, select **Programs > NetBackup > NetBackup Agents > 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 progress reports created for NetBackup for SQL Server on the client" on page 180.
- 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 125.
- 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 126.
- 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 127.

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 10

Restoring Backup Exec images with NetBackup for SQL

This chapter includes the following topics:

Using NetBackup to restore SQL Server from Backup Exec images

NetBackup can restore SQL Server from Backup Exec images, using the NetBackup Backup, Archive, and Restore (NetBackup Client) interface.

Requirements for restoring SQL Server from Backup Exec images

The following requirements apply when you restore SQL Server from Backup Exec images:

- To restore Backup Exec images successfully, you need to do the following: Import the Backup Exec media into NetBackup with the `vmphyinv` command, run the `bpimport` command, Phase 1, and Phase 2. Refer to the instructions for importing images from Backup Exec media.
  See the NetBackup Administrator’s Guide, Volume I.
  NetBackup can restore Backup Exec images of SQL Server.
- You must grant NetBackup permissions to perform backups and restores of SQL Server.
  See “Configuring the NetBackup services for SQL Server backups and restores” on page 26.

Limitations when restoring SQL Server from Backup Exec images

The following limitation exist when you restore SQL Server from Backup Exec images:

- Multiple databases cannot be restored in a single restore job. You can restore full, differential, and transaction log backups of a same database in a single restore job.
- Table backups cannot be restored, though they are included in the backup image and can be selected for restore. If you try to restore a table, the job fails with the error "unsupported object was selected for restore".

Specifying the server, client, and policy type

To browse for backups of SQL databases that are backed up by Backup Exec, you must first specify the server that contains the backup images. You must also specify the client that performed the backups, and the type of policy that is associated with the backups.

To specify the server, client, and policy type

1. Select File > **Specify NetBackup Machines and Policy Type**.
2. In the **Specify NetBackup Machines and Policy Type** dialog box, select the desired server from the **Server to use for backups and restores** list.
3. From the **Source client for restores** list, select the desired client.

   The source client is the SQL Server machine name whose backup images you want to browse.
4. From the **Policy type for restores** list, select **MS-SQL-Server**.
5. To redirect a restore to a different client, select the client to which to redirect the restore from the **Destination clients for restores** list.

   See "Redirecting a restore that was backed up with Backup Exec" on page 138.
6. Click **OK**.

   NetBackup browses for SQL Server backup images.

   The NetBackup History pane displays SQL Server backup information. The top split windows show individual image information. The bottom split gives file and folder information and also allows the user to select what files are to be restored.

Restoring from Backup Exec images

When you restore SQL Servers from Backup Exec images, there are several options and procedures that you can use. For example, you can redirect a restore that you backed up with Backup Exec. You can also restore Backup Exec SQL transaction logs up to a point in time, and restart SQL using copies of the master databases. You can restore the SQL master database from a Backup Exec image.

Restoring Backup Exec SQL transaction logs up to a named transaction

When restoring SQL Server transaction logs, you can restore transactions from a transaction log up to and including a named transaction (or mark). After the named transaction is reached, recovery from the transaction log is stopped.

To restore from a Backup Exec image a SQL transaction log up to a named transaction

1. Log on as Administrator.
2. Open the Backup, Archive, and Restore interface.

3 Specify the appropriate server, client, and policy type.

See "Specifying the server, client, and policy type" on page 130.

4 **Select File > Select Files and Folders to Restore > from Backup Exec Backup**.

5 In the **Restore** window, from the NetBackup History pane, select the backup image that contains the objects you want to restore. In the **All Folders** pane, select the most recent full database backup. Select also the most recent differential database backup, if any. And select all the log backups you want to restore.

6 Select **Actions > Start Restore of Marked Files**.

7 In the **Restore Marked Files** dialog box, from the Recovery completion state group, select **Leave database operational. No additional transaction logs can be restored**.

8 Select **Restore log up to named transaction** and then enter the name of the transaction.

The names are case sensitive. Make sure that you enter the correct uppercase and lowercase characters.

Select from one of the following options:

- To include the named transaction in the restore, select **Include the named transaction**.
- To specify a particular named transaction in the log, select **Found after** and select a date and time.
  If a date and time are not entered, recovery from the transaction log is stopped at the first transaction with the specified name.

9 Click **Start Restore**.

Restore options for restoring SQL backups from Backup Exec images

Microsoft SQL Server Redirection tab field descriptions describe the restore options that are available when you restore SQL backups from Backup Exec images.

Table 10-1 Restore options for restoring SQL backups from Backup Exec images

Item	Description
Recovery completion state	■ **Leave database operational. No additional transaction logs can be restored**. (With Recovery) Select this option to restore the last database, differential, or log backup in the restore sequence. This action causes the restore operation to roll back all incompleted transactions. After the recovery operation, the database is ready for use. If Leave database operational is not performed, the database is left in an intermediate state and is not usable. If **Leave database operational** is selected when an intermediate backup is applied, you cannot continue to restore backups. You must restart the restore operation from the beginning.
	■ **Leave database nonoperational but able to restore additional transaction logs** (No Recovery) Select this option during a restore if you have additional differential or transaction log backups to be restored in another restore job.
	■ **Leave database read-only and able to restore additional transaction logs** (Standby) Select this option during transaction log and database restore to create and maintain a standby database. See your SQL documentation for information on standby databases.
Replace databases or filegroups	Select this check box to replace a database or filegroup, even if another database or filegroup with the same name already exists on the server. If **Replace Databases or Filegroups** is not specified for a restore, SQL performs a safety check. This check ensures that a different database or filegroup is not accidentally overwritten. Refer to your SQL documentation for more information about the safety check that occurs when the REPLACE option is not selected.

Table 10-1 Restore options for restoring SQL backups from Backup Exec images *(continued)*

Item	Description
Automate master database restore	Enable NetBackup to stop SQL so that the master database can be restored. All existing users are logged off, and SQL Server is put into single-user mode. When this option is selected, only the master database can be restored; if this option is selected for any other database, those jobs fail. NetBackup must have access to the following SQL registry keys: HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer Without this access, a restore to the default directory may not work. Also, the option **Automate master database restore** does not work. To ensure that NetBackup has access rights, verify that the user account of the NetBackup Client Service has administrator rights to the Windows server that the SQL instance is installed on.

Table 10-1 Restore options for restoring SQL backups from Backup Exec images *(continued)*

Item	Description
Consistency check after restore	When a set of restores is complete and you want to recover the database, you select the **Leave database operational** option. You must also select from the following consistency checks: ■ **Full check, excluding indexes**. Select this option to 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 nonclustered index pages is not checked. ■ **Full check, including indexes**. Select this option to include indexes in the consistency check. Any errors are logged. This option is selected by default. ■ **Physical check only (SQL 2000 only)**. Select this option to perform a low overhead check of the physical consistency of the SQL Server 2000 database. This option only checks the integrity of the physical structure of the following: the page and the record headers, the consistency between the pages' object ID and index ID, and the allocation structures. ■ **None**. Select this option to perform sequential restores. Do not run a consistency check after a restore until all sequential restores are done. If a consistency check is selected during a restore, the restore completes but the consistency check is not performed. Check the job log for this information.

Table 10-1 Restore options for restoring SQL backups from Backup Exec images *(continued)*

Item	Description
Alternate drive for restoring database files	Select a drive to which SQL database files can be restored. Use this option if the drive where one or more of the database files previously resided no longer exists. **Note:** Do not select the **Alternate drive** option when you restore filegroups. Filegroups must be restored to the same drive letter and path that they were backed up from. In the backup set, SQL Server stores the physical file names of the files that make up the database. The physical file name includes the directory path. For the logical file, "pubs," the physical file name is stored as `E:\`*`install_path`*`\pubs.mdf`. If the database is restored later, SQL uses these same physical file names for the target of the restore. During a restore, NetBackup automatically creates any necessary subdirectories that do not exist. However, the behavior is different if the drive where one or more of the database files previously resided no longer exists. Then NetBackup moves those files to their original directory path, but on the alternate drive specified. Using the same example, if drive C: is specified, then the file with the original directory path of E:\*`install_path`*`\pubs.mdf` is restored to C:\*`install_path`*`\pubs.mdf`\pubs.mdf. If no alternate drive is specified in this situation, the job fails.
Restore to alternate drive	You can restore to an alternate drive in one of the following ways: ■ **Only when original drive does not exist** Select this option to restore all database files to their original directory path on the alternate drive. The restore is performed only if the drive where they originally resided exists. To make this option available, select a drive letter in **Alternate drive for restoring database files** list.
	■ **Even when original drive does exist** Select this option to restore all database files to their original directory path on the alternate drive. The restore is performed even if the drive where they originally resided exists. To make this option available, select a drive letter in **Alternate drive for restoring database files** list.

Table 10-1 Restore options for restoring SQL backups from Backup Exec images *(continued)*

Item	Description
Restore all database files to the target instance's data location	Select this check box to restore files to the default data and log directories of the destination instance. For example, to restore a database to a different instance of SQL. This option moves the database files to the correct location for the new instance. **Note:** Do not select the **Restore all database files to the target instance's data location** option when restoring filegroups. Filegroups must be restored to the same drive letter and path that they were backed up from. If this option is not selected, then the files are restored to the directory that the master database is in.
Point in time log restore	Select this check box to restore transactions from a transaction log up to and including a point in time in the transaction log. After the point in time, recovery from the transaction log is stopped. Select the part of the date you want to change. Then enter a new date or click the arrow to display a calendar from which you can select a date. Select the part of the time you want to change, and then enter a new time or click the arrows to select a new time.
Restore log up to named transaction (SQL 2000)	Select this check box to restore transactions from a transaction log up to a named transaction (or named mark) in the transaction log. After the named transaction is restored, the recovery from the transaction log is stopped. The named transactions are case-sensitive.
	You can restore transactions in one of the following ways: ■ **Include the named transaction** Select this check box to include the named transaction in the restore; otherwise the restore stops immediately before the named transaction is restored.
	■ **Found after** Select this check box to specify a date and time after which the restore operation is to search for the named transaction. For example, consider that you specified a restore from a log up to the named transaction "AfternoonBreak." This transaction was found after 6/02/2012, 12:01 P.M. In that case, the restore operation does not search for "AfternoonBreak" until after that time.

Redirecting a restore that was backed up with Backup Exec

You can redirect the following Backup Exec backups:

- A database backup to a different server, database, or instance.
- Differential and log backups to wherever the associated database is restored.
- One or more filegroups in a backup to a different server or instance. Filegroups can be redirected to a different server, but the database file paths cannot be changed. For example, if the filegroup was backed up from `G:\SQLDATA`, it must be restored to `G:\SQLDATA` even if it is redirected to another server. Filegroups must be restored to the same drive letter and path that they were backed up from.

To redirect a restore that was backed up with Backup Exec

1. Log on as Administrator.
2. Open the Backup, Archive, and Restore interface.
3. Specify the appropriate server, source client, destination client, and policy type.

   See "Specifying the server, client, and policy type" on page 130.
4. Select File > **Select Files and Folders to Restore > from Backup Exec Backup**.
5. In the **Restore** window, select the objects you want to restore.
6. Select Actions > **Start Restore of Marked Files**.
7. In the **Restore Marked Files** dialog box, click the Microsoft SQL Server Redirection tab.
8. Select the other restore options that you want on the **Microsoft SQL Server** tab.
9. Click **OK**.

Restoring Backup Exec SQL transaction logs backups up to a point in time

You can restore transactions from a transaction log up to and including a point in time in the transaction log. After the point in time is reached, recovery from the transaction log stops.

To restore SQL transaction logs up to a point in time from Backup Exec images

1. Log on as Administrator.
2. Open the Backup, Archive, and Restore interface.
3. Specify the appropriate server, client, and policy type.

   See “Specifying the server, client, and policy type” on page 130.
4. Select File > **Select Files and Folders to Restore > from Backup Exec Backup**.
5. In the **Restore** window, from the NetBackup History pane, select the backup image that contains the objects you want to restore.
6. In the **All Folders** pane, select the most recent full database backup. Select also the most recent differential database backup, if any. And select all the log backups you want to restore.
7. Select Actions > **Start Restore of Marked Files**.
8. In the **Restore Marked Files** dialog box, from the Recovery completion state group, select **Leave database operational. No additional transaction logs can be restored**.
9. Select **Point in time log restore** and select a date and time.
10. Click **Start Restore**.

Restoring the SQL master database from a Backup Exec image

If the master database is damaged, symptoms can include the following:

- An inability to start SQL.
- Segmentation faults or input or output errors.
- A report that is generated by SQL Database Consistency Checker utility (DBCC).

If you can start SQL, you can restore the latest copy of the master database backup by using the **Automate master database restore** option. Then restore any other databases, if needed.

If you cannot start SQL, you can use copies of the master database and model databases.

See “Restarting SQL using database copies” on page 140.

When a backup is restored, all changes that are made to the master database after the last backup was created are lost. You must reapply any changes. If you created any user databases after the master database was backed up, you cannot access those databases. You cannot access the databases until the databases are restored from backups or reattached to SQL.

To restore the SQL master database from a Backup Exec image

1. Log on as Administrator.
2. Open the Backup, Archive, and Restore interface.
3. Specify the appropriate server, client, and policy type.

   See "Specifying the server, client, and policy type" on page 130.
4. Select File > **Select Files and Folders to Restore > from Backup Exec Backup**.
5. In the **Restore** window, from the NetBackup History pane, select the backup image that contains the last master database backup.
6. In the **All Folders** pane, select the master database.
7. Select Actions > **Start Restore of Marked Files**.
8. In the **Restore Marked Files** dialog box, on the **Microsoft SQL Server** tab, select **Automate master database restore**.

   When this option is selected, only the master database can be restored; if this option is selected for any other database, those jobs fail. When you start the restore operation, all existing users are logged off, and SQL Server is put into single-user mode.

   If NetBackup does not have access to the SQL registry keys `HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server` and `HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer`, then a restore to the default directory may not work. And the option **Automate master database restore** does not work. To ensure that NetBackup has access rights, verify that the account that NetBackup uses has administrator rights to the computer that is running SQL.
9. Select a consistency check to be run after the restore.
10. Click **Start Restore**.

    After the restore, SQL is restarted in multi-user mode.

Restarting SQL using database copies

If the master database is critically damaged and SQL cannot be started, you can: replace the corrupted or missing databases with the copies of the master databases and model databases. Backup Exec automatically creates and updates these copies whenever backups of those databases are run. (Alternatively, restart SQL by running the Rebuild Master utility or reinstall SQL.)

After SQL is running again, you can restore the latest copy of the master database and restore any other databases, if needed.

See "Restoring the SQL master database from a Backup Exec image" on page 139.

If copies of the master databases and model databases were not made, then you must use Microsoft's `rebuildm.exe` utility to rebuild the master database and start SQL.

To restart SQL using database copies

1. Verify that the database copies are present.

   The database copies are named `master$4idr`, `mastlog$4idr`, `model$4idr`, and `modellog$4idr`.

   If necessary, restore the master database and model database copies from a backup set to the same directory. Restore them from the original master and model databases are in.

2. Open a command prompt window and delete the original master and model databases and their transaction logs. For example:

   ```
   del master.mdf mastlog.ldf model.mdf modellog.ldf
   ```

3. Rename the copies of the databases back to their original names. Type the following:

   ```
   rename master$4idr master.mdf
   mastlog$4idr mastlog.ldf
   model$4idr model.mdf
   modellog$4idr modellog.ldf
   ```

4. Use the SQL Service Control Manager to start SQL Server.

5. Continue with the procedure to restore the latest changes to the master database.

   See "Restoring the SQL master database from a Backup Exec image" on page 139.

Restoring Backup Exec filegroup backups

With filegroup backups, you can restore the entire database, a primary filegroup, a filegroup that contains a deleted or changed table, and a nonprimary filegroup.

Note: Use separate restore jobs to restore the primary filegroup, the rest of the filegroup backup sets, and the transaction logs.

To restore a SQL filegroup from a Backup Exec image

1. Log on as Administrator.
2. Open the Backup, Archive, and Restore interface.
3. Specify the appropriate server, client, and policy type.

   See "Specifying the server, client, and policy type" on page 130.
4. Select File > **Select Files and Folders to Restore > from Backup Exec Backup**.
5. In the **Restore** window, from the NetBackup History pane, select the backup image that contains the objects you want to restore.
6. In the **All Folders** pane, select the full backup of the primary filegroup and any differential backups.
7. Select Actions > **Start Restore of Marked Files**.
8. In the **Restore Marked Files** dialog box, from the Recovery completion state group, select **Leave database nonoperational but able to restore additional transaction logs**.
9. Click **Start Restore**.
10. After the primary filegroup is restored, select the latest full and differential backups for the other filegroups.
11. On the **Microsoft SQL Server** tab, select the recovery completion state **Leave database nonoperational but able to restore additional transaction logs**, and then start the restore job.
12. When the other filegroups are restored, select the transaction logs.
13. On the **Microsoft SQL Server** tab, for the recovery completion state, select **Leave database operational and restore all of the transaction logs**.
14. Optionally, you can also select **Point in time log restore** or **Restore log up to named transaction**.
15. Click **Start Restore**.

Restoring Backup Exec Database backups

If the database you want to restore uses the simple recovery model, there are no transaction log backups to restore. You only need to restore the most recent full database backup and if you were running differential database backups, restore the most recent differential database backup.

To restore a SQL database from a Backup Exec backup

1. Log on as Administrator.
2. Open the Backup, Archive, and Restore interface.
3. Specify the appropriate server, client, and policy type.

   See "Specifying the server, client, and policy type" on page 130.
4. Select File > **Select Files and Folders to Restore > from Backup Exec Backup**.
5. In the **Restore** window, in the **NetBackup History** pane, select the backup image that contains the objects you want to restore.
6. Select the most recent full database backup and the most recent differential database backup to restore.
7. Select Actions > **Start Restore of Marked Files**.
8. In the **Restore Marked Files** dialog box, from the Recovery completion state group, select **Leave database operational. No additional transaction logs can be restored**.
9. Click **Start Restore**.

Chapter 11

Using NetBackup for SQL Server in SQL Server High Availability (HA) environments

This chapter includes the following topics:

About SQL Server High Availability environments

Microsoft SQL Server supports the following types of HA environments: SQL Server clusters, database mirroring, and log-shipping. For complete descriptions of these HA solutions refer to *SQL Server Books Online*. Each solution uses log porting to synchronize one or more copies of selected databases with databases on alternate SQL Server installations. A manual or an automatic failover results in continued access for mission critical database applications.

Each of the solutions requires a *primary* instantiation of a database and one or more partner or secondary copies of the database. Each solution requires that the relationships between the primary and the secondary instantiations are defined and configured as one of these solution types, with varying options. Normally, the instantiations are on different hosts but have the same database name. When the environment is active, SQL Server automatically ensures that the secondary instantiations are synchronized with the primary instantiation. The secondaries serve as hot standbys to the primary because the primary is where all database transactions take place. Depending on the type of HA solution used, failover occurs automatically or manually and a secondary can then serve as the new primary copy.

A brief comparison between these HA environments is as follows:

- SQL Server Clusters
  - A single SQL Server instance can have multiple synchronized copies with the same name on different hosts.
  - System and user databases are stored on shared storage.
  - The primary server is called the *active* node, while the secondary servers are called the *inactive* nodes.
- Database mirroring
  - A single database can have a single synchronized copy with the same name on the same or on a different host.
  - The primary database is called the *principal*, while the secondary database is called the *mirror*.
  - As of SQL 2012, Microsoft has deprecated this HA environment.
- Log-shipping
  A single database can have multiple synchronized copies with the same name on the same or on different hosts.

About using NetBackup for SQL in a cluster

NetBackup for SQL Server can be used to back up and to restore any installations that are clustered. NetBackup for SQL Server is supported with Microsoft Cluster server (MSCS) or Veritas Cluster Server (VCS).

Note the following when you use NetBackup for SQL in a cluster:

- For VCS clusters, the `VirtualName` attribute under the Veritas Cluster Server resource type, `Lanman`, is the name of the virtual SQL Server.
- For MSCS clusters, the virtual server name distinguishes the unique SQL Server instances.
- You perform a backup or restore with a clustered instance of SQL Server as with a non-clustered instance with two exceptions. The client name or host name

is the SQL virtual name or `VirtualName` and the designated instance is `<default>`.

Configuring the NetBackup server to be aware of clustered SQL Server instances

Perform the following configuration steps after the installation of a virtual SQL Server (VIRTUALSERVER) has been created. Also perform the installation of a virtual NetBackup media server. The following actions must be performed on the master server or on a NetBackup remote client console that acts for the master server.

To configure NetBackup server to be aware of clustered SQL Server instances

1 Using the NetBackup Administration Console, create a MS-SQL-Server policy (for example, VIRTSQLPOLICY), to specify the storage attributes of the backup. Set up the following attributes:

- Policy-storage unit: Specify a storage unit that belongs to the storage group that is intended for backup. If you use a virtual media server, then specify a storage unit that belongs to the virtual media server.
- Create a backup policy schedule for VIRTSQLPOLICY.
- Add the virtual SQL Server name (VIRTUALSERVER) to the client list.

2 To incorporate SQL Server backup scheduling, create an automatic schedule in the VIRTSQLPOLICY. Create the automatic schedule as follows:

- Add an automatic backup schedule to VIRTSQLPOLICY.
- Add one or more script names (batch files) to the file list.

3 Create a standard backup policy (say, STDPOLICY). Add all physical names in the cluster to the client list of STDPOLICY.

4 Establish the permissions settings as follows for a redirected restore to a different client.

On the master server, do one of the following:

- Create a *install_path*`\NetBackup\db\altnames\No.Restrictions` file.
- Create each of the files, *install_path*`\NetBackup\db\altnames\NODEA`, *install_path*`\NetBackup\db\altnames\NODEB`.

Note: The `No.Restrictions` file allows all clients to perform redirected restores to different clients. This parameter may need to be added and removed according to site policies.

Performing a backup on a virtual SQL Server instance

NetBackup for SQL Server supports browsing for databases and images on any of the physical nodes. But a user backup or restore is not successful unless it is launched from the NetBackup client that is active. After NetBackup has been properly configured: a backup of a SQL Server database from a virtual instance is similar to a backup of a database from a non-virtual one.

To perform a backup on a virtual SQL Server instance

1. Open the NetBackup for SQL Server interface on the active platform node.
2. Select **File** > **Backup SQL Server objects**.
3. Select one or more databases.
4. In the **NetBackup Policy** field, enter the name of the MS-SQL Server policy (VIRTSQLPOLICY).This policy was created for specifying storage attributes for the virtual SQL Server backup.
5. Click **OK**.

Performing a restore on a virtual SQL Server instance

This procedure describes how to perform a restore on a virtual SQL Server instance.

To perform a restore on a virtual SQL Server instance

1. Open the NetBackup for SQL Server interface on the active platform 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) as the SQL host.
4. From the **SQL Server Instance** drop-down list, select **<default>**.
5. Click **OK**.
6. In the **Restore Microsoft SQL Server Objects** dialog box, select a backup image or staged image list.
7. Click **OK**.

About using SQL Server in a cluster with a multi-interface network connection (multi-NIC)

To combine SQL Server clusters with the use of public-private interfaces to perform backups, special configuration is required. Many administrators want to reserve a separate network interface for their SQL Server host machines that is used for routing backup traffic. This type of configuration requires special consideration in terms of configuring both the NetBackup master server and the NetBackup client which backs up SQL Server. In addition, a special consideration is required in terms of how to browse for SQL Server backup images.

The following distinct network resources can be identified in a dual-interface SQL Server cluster.:

- 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 public virtual name of the SQL Server (for example, VIRTSQL), and
- The private virtual name of the SQL Server (for example, VIRTSQL-NB).

Configuring the master server for a SQL cluster with a multi-NIC

The following configuration changes must be made on the master server to allow for backups and restores over a private interface:

- The backup policies must include the private interface name in the clients list of the policy. The NetBIOS or public name of the client should not be used..
- Permissions must be added to allow for browsing of backups across the private interface. The administrator can allow all clients or allow single clients to browse and restore a backup that is performed over the multi-NIC connection.

To configure the master server for a SQL Server in a cluster with a multi-NIC

1. Open the NetBackup Administration Console.
2. Create a new policy or open an existing policy.
3. In the clients list for the policy, add a new client.

   Specify the client name as the private SQL virtual instance name. For example, VIRTSQL-NB.
4. Configure one of the following:
   - Allow any NetBackup client to browse the backups the SQL Server host machine makes.
     Add the empty file `NB_INSTALL\db\altnames\No.Restrictions` on the master server.
   - Allow only the SQL Server host machine, SQLHOST, to access the backups the master server manages.
     Add the empty file, `NB_INSTALL\db\altnames\`*SQLHOST* on the master server.

Specifying the private name of the SQL client

To configure backups of a client in a multi-NIC environment, you must specify the private interface name of the SQL Server host machine. You can set the Client name during the installation of the NetBackup client. Or it can be changed through the Backup, Archive, and Restore interface on the client, in the **NetBackup Client Properties** dialog box.

To specify the private name of the client

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 would be SQLHOST1-NB. The private name for the computer SQLHOST2 would be SQLHOST2-NB.

Performing backups in a SQL cluster with a multi-NIC connection

To perform backups of SQL Server using a multi-NIC connection, you need to specify the public name of the virtual SQL Server host. You also need to create and edit a batch file for backups so that it includes the private virtual SQL Server name.

To perform backups in a SQL cluster with a multi-NIC connection

1. On either SQLHOST1 or SQLHOST2, 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 associated with the BROWSECLIENT from the public virtual SQL Server name 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 both nodes in the cluster so that it is available for scheduled backups. Backups are done regardless of which node is active when a backup is initiated.

Performing restores in a SQL cluster with a multi-NIC connection

To perform restores of SQL Server in a multi-NIC, cluster environment, you need to do the following:

- Specify the public name of the virtual SQL Server host. (In the **SQL Server connection properties** dialog box.)
- Indicate both the virtual SQL Server host name and the private interface name of the virtual SQL Server host. (In the **Browse History Options** dialog box.)
- Create a batch file for restores and manually edit it to include the private name of the virtual SQL Server.

To perform restores in a cluster with a multi-NIC connection

1. On either SQLHOST1 or SQLHOST2, 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, do the following.

SQL Host	Specify the public name of the virtual SQL Server (VIRTSQL).
Source Client	Specify the 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 72.

   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**.

    A batch file similar to the following is generated.

```
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 virtual SQL Server name 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**.

About NetBackup for SQL Server with database mirroring

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 200x 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 153.
3. Permission for a redirected restore must be established for both mirroring partners. Do one of the following:
   - On the master server, create a file called:
     `install_path\NetBackup\db\altnames\No.Restrictions`

     Note: When you create the `No.Restrictions` file it allows all clients to perform redirected restores to different clients. This file may need to be added and removed according to your site policies.
   - Create each of the files on the master server:

     ```
     install_path\NetBackup\db\altnames\hostA
     install_path\NetBackup\db\altnames\hostB
     ```

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 SQL Client.

When you restore a mirror database, you must run the NetBackup 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

To use log-shipping with NetBackup, both the primary and the secondary should be set up as clients of the same master server.

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 156.

3 Permission for a redirected restore must be established for both the primary and the secondary. Do one of the following:

- On the master server, create a file called:
  install_path`\NetBackup\db\altnames\No.Restrictions`

Note: When you create the `No.Restrictions` file it allows all clients to perform redirected restores to different clients. This file may need to be added and removed according to your site policies.

- Create each of the files on the master server:

```
install_path\NetBackup\db\altnames\hostA
      install_path\NetBackup\db\altnames\hostB
```

Backing up SQL Server in an environment with 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.

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, the result cannot be predicted. However, you may 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 12

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 Microsoft 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. Select File > **Set SQL Server connection** properties.

See “About SQL differential backups” on page 160.

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 121.

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 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

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 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 Filegroups 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.

Partial Database Backup

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.

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.

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.

About SQL 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 12-1 describes the components of NetBackup for SQL Server.

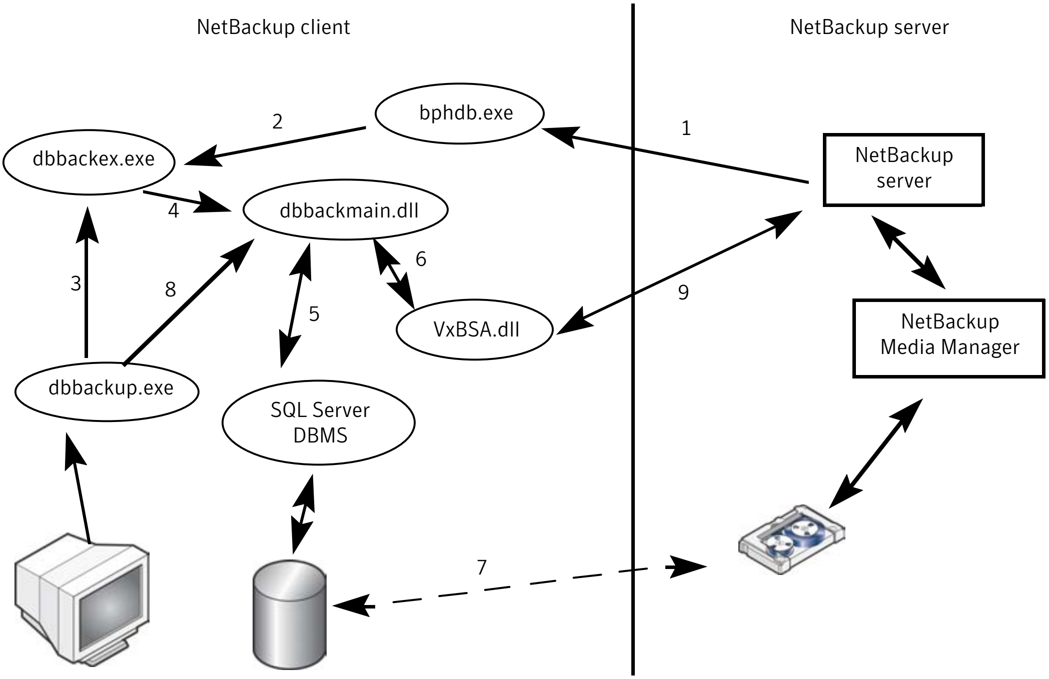
Table 12-1 Components of NetBackup for SQL Server

Component	Filename	Description
graphical user interface (GUI)	dbbackup.exe	You use this interface to browse database objects and backup images, create backup and restore scripts, and launch backup and restore 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 12-1 shows the relationships of NetBackup for SQL Server with other software components.

Figure 12-1 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.

- 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 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. However, a clustered instance of SQL Server resides jointly on multiple hosts and is identified with a virtual name.

Backups for a host are cataloged under the client name that is specified in the client properties for that host. Ordinarily, the client name is the same as the host name on which SQL Server resides. Normally, this name is the host's NetBIOS name (such as TIGER). However, some backup administrators prefer to identify the client to NetBackup in other ways. For example, an administrator might use the IP name (for example, 20.81.74.123) or the fully qualified domain name (tiger.apexworks.com). Or, an administrator might use an alternate network interface that is attached to the host (tiger1.apexworks.com). For clustered SQL instances, the client name is the virtual SQL Server name that is used for the virtual SQL Server installation.

In most cases when you browse for backup images using the NetBackup Microsoft SQL Client, you only need to specify the SQL Server host name. Specify this name 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:

- Retrieval of backups on a network interface not having 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, you should specify the host name in the **SQL Host** box and the network interface name in the **Source Client** box.
SQL Host: TIGER
Source Client: TIGER1

- Retrieving 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.
  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.
  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
- Retrieval of the backups that were backed up from a SQL Server cluster.
  This scenario does not present any special problem because the images are stored under the cluster name. Provide the **SQL Host name** as the virtual name and use the default value for the **Source Client**.

How does NetBackup for SQL Server back up a database?

From the NetBackup Microsoft SQL Client you can select the databases or the database components that you want to back up. When you select an object for backup, the GUI creates a batch file that contains a script. The SQL Server database agent interprets this script. You can either execute the script immediately or save it to run later. You can run a script later by including it in the backup selections list of a backup policy.

When a backup script is executed, NetBackup for SQL Server does the following: generates an SQL 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 Microsoft 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. An SQL 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 NetBackup 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 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.

When you back up a file or filegroup, 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 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 on-line 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.

To reduce backup size and time by with read-only filegroups

1. Create a policy for read-only filegroups.
2. Create an Application Backup schedule with Retention set to infinite.
3. Create a policy for read-write filegroups.
4. Create an Automatic Backup schedule with the wanted Retention period and add the read-write filegroups batch file to the **Backup Selections** list.
5. Back up read-only filegroups.
6. In the NetBackup for SQL Server interface, select the policy you created in step 1 and perform this backup one time.

7. All read-only filegroups must be included in some combination of full, partial, or individual filegroup and file backups.

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

   ---

8. Back up read-write filegroups.
9. This backup is done automatically through the schedule you created in step 3.
10. Or you can manually back up the read-write filegroups.
11. View the read-only backup set.

    If necessary, confirm all read-only groups are backed up.

What factors affect the data transfer rate during a SQL Server backup or restore operation?

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. With this information as a background, you still must adjust several things specific to SQL Server. Note that some of the following factors are only applicable to SQL Server stream-based operations and have no affect on snapshot backups or restores. A detailed description of how to make these adjustments is available.

See “About NetBackup for SQL performance factors” on page 52.

The following factors affect the data transfer rate during a SQL Server backup or restore operation.

`MAXTRANSFERSIZE`	(stream-based only) The size that SQL Server uses for transferring data. Among the SQL Server-specific factors, `MAXTRANSFERSIZE` has the greatest overall potential for optimizing transfer rates. In general, improved performance correlates directly with an increased `MAXTRANSFERSIZE` value.

`NUMBER OF BUFFERS PER STRIPE`

(stream-based only) Setting this factor to a value greater than one enables multi-buffer during data transfer. 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.

This setting corresponds to the `NUMBUFS` keyword. You can also use the `BUFFERS` keyword.

`NUMBER OF STRIPES`

You can improve performance if you increase the number of stripes that you use to back up a SQL Server object. You must meet the following requirements:

- Each stripe is backed up to a separate tape unit (the stripes are not multiplexed to the same drive).
- You have reached the maximum transfer rate of any single drive.

Performance varies depending on your environment. Since each stripe uses individual threads, multiple stripes may lead to contention when they access internal SQL Server structures. A cost that is associated with management of the threads. However, you may see improvement on a large system if the cost to start starting and stop I/O is greater than using additional stripes.

`SHARED MEMORY`

Shared memory should always be used if it is available in your backup environment.

`ALTERNATE BACKUP METHOD`

(stream-based only) A transfer method in which SQL Server and NetBackup are allowed to share the same data transfer buffers. This method may not be faster than ordinary shared memory data transfer. But this method results in decreased CPU utilization because it avoids a data copy step. If you use `ALTERNATE BACKUP METHOD`, Symantec recommends that you have `MAXTRANSFERSIZE` set to the maximum value.

About recovery factors for SQL Server

Take into consideration the following factors when you define a recovery plan for your application environment. Much of this information is based on Microsoft's *SQL Server Books Online*. See that resource for a more inclusive discussion.

Transaction logs	See “About SQL Server transaction logs” on page 169. See “About backing up the transaction log” on page 170.
Recovery and recovery strategies	See “About recovery strategies” on page 169. See “About database recovery” on page 172. See “About staging recovery” on page 172.
Differential backups	See “About differential backups” on page 170.
File and filegroup backups	See “About file and filegroup backups” on page 171.

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 169.

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 on an hourly basis.

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 172.

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 as frequently as every hour 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 166.

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 72.

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 72.

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 following backup types:

- Database
- Database differential
- Filegroup
- Filegroup differential
- File

- Read-only filegroups
- Read-only filegroups differential
- Partial database
- Partial database differential
- Transaction log

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.

Performing user-directed operations with dbbackex

This chapter includes the following topics:

- Using dbbackex to perform user-directed operations for SQL Server
- Using client-based schedulers with dbbackex

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 38.

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 57.

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
- About NetBackup for SQL Server backup names

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.

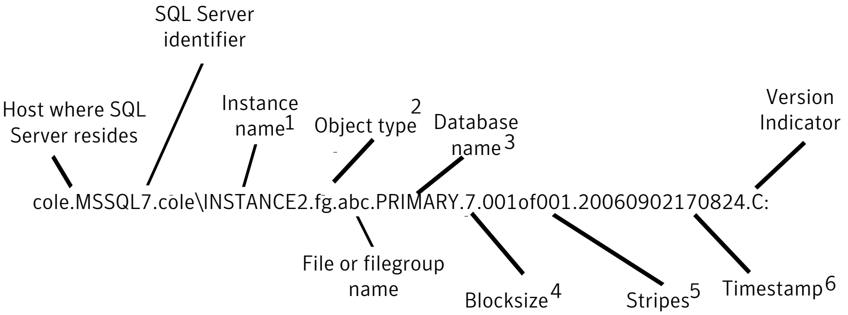
The following example shows how to obtain the list of SQL Server backups that were backed up from client juneberry to server Cole:

```
C:\Program Files\NetBackup\bin\bplist -C juneberry -t 15 -S cole -R \
juneberry.MSSQL7.JUNEBERRY.db.pubs.~.7.001of003.20060920101716..C:\
juneberry.MSSQL7.JUNEBERRY.db.pubs.~.7.002of003.20060920101716..C:\
juneberry.MSSQL7.JUNEBERRY.db.pubs.~.7.003of003.20060920101716..C:\
juneberry.MSSQL7.JUNEBERRY.fil.pubs.pubsnew.7.001of001.20060919175149..C:\
juneberry.MSSQL7.JUNEBERRY\NEWINSTANCE.trx.abc.~.7.001of001.20060902170920..C:\
juneberry.MSSQL7.JUNEBERRY\NEWINSTANCE.fg.abc.PRIMARY.7.001of001.20060902170824.C:\
juneberry.MSSQL7.JUNEBERRY\NEWINSTANCE.db.Howard's
Barbeque.~.7.001of001.20060901085255..C:\
juneberry.MSSQL7.JUNEBERRY\NEWINSTANCE.inc.Howard's
Barbeque.~.7.001of001.20060903108552..C:\
juneberry.MSSQL7.COLE.db.pubs.~.7.001of001.20060907100101..C:\
juneberry.MSSQL7.COLE.db.pubs.~.7.001of001.20060908200234..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.



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
fdg	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 blocksize is specified as 256 kilobytes bytes * 2 ^{blocksize}.

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*.

7 - The character that immediately precedes the version indicator is the delimiter, which by default is a period. However, if a period is used in any of the fields, the delimiter may be another character. "C" is for NetBackup 6.0 and higher.

Troubleshooting

This chapter includes the following topics:

About progress reports created for NetBackup for SQL Server on the client

NetBackup for SQL Server creates a progress report for each operation that has been initiated. This report contains summary information about the overall status of your job. The reports are contained in directory `*install_path*\NetBackup\logs\user_ops\MsSql\logs` and can be viewed by choosing File > **View status** in the NetBackup Database Client graphical user interface .

The progress report contains the following types of information:

- The batch keywords and values which define the operation
  Information about the batch file syntax is available.
- 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

Sample progress report for NetBackup for SQL Server backup

Table 15-1 shows a typical progress report that is created for a database backup.

Table 15-1 Progress report for NetBackup for SQL Server backup

Line	Text
1	OPERATION backup
2	Database "DatabaseA"
3	SQLHOST "JUY"
4	SQLINSTANCE "NEWINSTANCE"
5	NBSERVER "JUY"
6	MAXTRANSFERSIZE 0
7	BLOCKSIZE 0
8	ENDOPER TRUE
9	INF - BACKUP started using
10	Microsoft SQL Server 2012 - 11.0.2100.60 (X64)
11	Feb 10 2012 19:39:15
12	Copyright (c) Microsoft Corporation
13	Enterprise Edition (64-bit) on Windows NT 6.1 <X64> (Build 7601: Service Pack 1) (Hypervisor)

Table 15-1 Progress report for NetBackup for SQL Server backup *(continued)*

Line	Text
14	Batch = C:\Program Files\Veritas\NetBackup\dbext\mssql\temp\__09_42_24_076_00.bch, Op# = 1
15	INF - Using backup image juy.MSSQL7.JUY\NEWINSTANCE.db.DatabaseA.~.7.001of001.20060701094227..C
16	INF - backup database "DatabaseA" to VIRTUAL_DEVICE='VNBU0-5652-1224-1120228947' with stats = 10, blocksize = 65536, maxtransfersize = 65536, buffercount = 1
17	INF - Number of stripes: 1, Number of buffers per stripe 1.
18	12:41:07 Initiating backup
19	12:41:10 INF - Starting bpbrm
20	12:41:12 INF - Data socket = juy.domain.com.4146
21	12:41:12 INF - Name socket = juy.domain.com.1527
22	12:41:12 INF - Job ID = 143
23	12:41:12 INF - Backup ID = juy_1058982070
24	12:41:12 INF - Backup time = 1058982070
25	12:41:12 INF - Policy name = SQL
26	12:41:12 INF - Snapshot = 0
27	12:41:12 INF - Frozen image = 0
28	12:41:12 INF - Backup copy = 0
29	12:41:12 INF - Master server = juy
30	12:41:12 INF - Media server = juy
31	12:41:12 INF - Multiplexing = 0
32	12:41:12 INF - New data socket = juy.domain.com.1388
33	12:41:12 INF - Use shared memory = 1
34	12:41:12 INF - Compression = 0
35	12:41:12 INF - Encrypt = 0

Table 15-1 Progress report for NetBackup for SQL Server backup *(continued)*

Line	Text
36	`12:41:12` INF - Client read timeout = 300
37	`12:41:12` INF - Media mount timeout = 0
38	`12:41:16` INF - Data buffer size = 262144
39	`12:41:18` INF - Beginning backup on server `juy` of client `juy`
40	INF - Thread has been closed for stripe #0
41	`12:41:25` INF - Server status = 0
42	`12:41:26` INF - Backup by `hao` on client `juy` using policy SQL: the requested operation was successfully completed.

Observe the following:

- Lines 1 to 8 contain the batch syntax which drove this operation. This operation was started from a batch file that the GUI created.
- Line 9 indicates that the operation was a backup ("Dump"). The term ‘Load’ would indicate a restore.
- Lines 10-13 provide the version of SQL Server and the Windows operating system.
- Line 14 provides the name of the batch file. Since the backup was launched immediately, it was placed in *install_path*\NetBackup\DbExt\MsSql\Temp directory.
- Line 16 provides the actual SQL syntax that was used to launch the backup command.
- Line 17 indicates that this backup was a single stream backup.
- Lines 18 to 42 provide NetBackup-based client parameters and statuses. Note that the server status on line 40 indicates that the operation was completed with status 0, which indicates success.

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.

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 184.

See "About restore operation debug logging for SQL Server" on page 184.

About backup operation debug logging for SQL Server

The following logs pertain to back up operations:

`install_path\NetBackup\logs\bphdb` (scheduled backups only)

`install_path\NetBackup\logs\dbclient`

`install_path\NetBackup\logs\bpbkar` (Snapshot Client)

`install_path\NetBackup\logs\bpfis` (Snapshot Client)

Log names are formatted as `mmddyy.log`, where *mm* is the month, *dd* is the day, and *yy* is the year.

About restore operation debug logging for SQL Server

The following logs pertain to restore operations:

`install_path\NetBackup\logs\dbclient`

`install_path\NetBackup\logs\bpbkar` (Snapshot Client) `install_path\NetBackup\logs\bpfis` (Snapshot Client)

`install_path\NetBackup\logs\bppfi` (Instant Recovery)

Log names are formatted as `mmddyy.log`, where *mm* is the month, *dd* is the day, and *yy* is the year.

Setting the maximum trace level for NetBackup for SQL Server

You can set the maximum trace level in the NetBackup SQL Client GUI or in the batch file.

To set the maximum trace level in the NetBackup SQL Client GUI

1. Open the NetBackup 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 batch file

1. Open the NetBackup 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.

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 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 52.

Troubleshooting VMware backups and restores of SQL Server

One Application State Capture job is created per VM, regardless of which applications are selected in policy.

Note the following when you perform a VMware backup that protects an application:

- 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 VMselection 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 Symantec nor the VMware VSS Provider is installed at the time of backup, the SQL databases are not quiesced. In this case, the recovery of a SQL 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.symantec.com/docs/TECH198864

Appendix A

Sample batch files

This appendix includes the following topics:

About NetBackup for SQL sample batch files

You can use batch files to initiate backup and restore operations. The batch file examples show you how to perform a variety of tasks with NetBackup for SQL Server batch files.

The following samples of batch files are available:

- See “NetBackup for SQL Server script to back up a database” on page 189.
- See “ NetBackup for SQL Server script to restore a database” on page 190.
- See “ NetBackup for SQL Server script to perform a striped database backup and allow multiple internal buffers per stripe” on page 190.
- See “NetBackup for SQL Server script to restore a database from multiple stripes” on page 191.
- See “ NetBackup for SQL Server script to restore a database transaction log up to a point in time” on page 191.
- See “NetBackup for SQL Server script to perform an operation and specify the user ID and password to use to SQL Server” on page 192.
- See “NetBackup for SQL Server script to perform multiple operations in sequence” on page 193.
- See “NetBackup for SQL Server script to perform a set of operations in parallel” on page 194.
- See “NetBackup for SQL Server script to specify the maximum transfer size and block size for a backup” on page 195.
- See “NetBackup for SQL Server script to stage a database restore from a database backup, a differential backup, and a series of transaction backups” on page 196.
- See “NetBackup for SQL Server script to stage a database restore from a filegroup backup, several file backups, and transaction log backups” on page 198.
- See “ NetBackup for SQL Server script that uses environment variables to exclude instances and databases from backup” on page 201.

NetBackup for SQL Server 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
```

NetBackup for SQL Server 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.~.7.001of001.20060628123631..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 177.

```
OPERATION RESTORE
OBJECTTYPE DATABASE
DATABASE "pubs"
# The following image is type: Full
NBIMAGE "cadoo.MSSQL7.CADOO\SECOND.db.pubs.~.7.001of001.20060628123631..C"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
BROWSECLIENT "CADOO"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE RECOVERED
ENDOPER TRUE
```

NetBackup for SQL Server 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
```

NetBackup for SQL Server 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.~.0.001of004.20060216151937..C
STRIPES 004
MAXTRANSFERSIZE 6
BLOCKSIZE 7
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
BROWSECLIENT "CADOO"
RECOVEREDSTATE RECOVERED
ENDOPER TRUE
```

NetBackup for SQL Server 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, 2006 at `2:03:00` P.M.). This time precedes the date of the backup log (Feb 16, 2006 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 20060216/14:03:00
DATABASE Northwind
NBIMAGE cadoo.MSSQL7.CADOO.trx.Northwind.~.0.001of001.20060216140321..C
MAXTRANSFERSIZE 6
BLOCKSIZE 7
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
BROWSECLIENT "CADOO"
ENDOPER TRUE
```

NetBackup for SQL Server 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 and NetBackup” on page 25.

```
OPERATION BACKUP
DATABASE "BUSINESS"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
NBSERVER "CHISEL"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
USERID JSMITH
PASSWORD my.Pwd
ENDOPER TRUE
```

NetBackup for SQL Server 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
```

NetBackup for SQL Server script to perform a set of operations in parallel

This sample is identical to the following sample except that the first operation contains `BATCHSIZE 3`.

See “NetBackup for SQL Server script to perform multiple operations in sequence” on page 193.

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
```

NetBackup for SQL Server 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 bytes * 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
```

NetBackup for SQL Server 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.~.7.001of001.20060701094227..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.~.7.001of001.20060701103323..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.~.7.001of001.20060701090005..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.~.7.001of001.20060701100030..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.~.7.001of001.20060701110015..C"
SQLHOST "CADOO"
SQLINSTANCE "SECOND"
```

```
NBSERVER "BOW"
BROWSECLIENT "CADOO"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE NOTRECOVERED
ENDOPER TRUE
```

NetBackup for SQL Server 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.7.001of001.20060701095634..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_FG2.7.001of001.20060701095425..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.
  20060701100824..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.
  20060701100908..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.
  20060701100953..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.~.7.001of001.20060701100030..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.~.7.001of001.20060701110015..C"
SQLHOST "CA"
SQLINSTANCE "SECOND"
NBSERVER "BOW"
BROWSECLIENT "CA"
MAXTRANSFERSIZE 6
BLOCKSIZE 7
RESTOREOPTION REPLACE
RECOVEREDSTATE RECOVERED
ENDOPER TRUE
```

NetBackup for SQL Server 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.

Index