Arcserve® Backup for Windows

Agent for Microsoft SQL Server Guide
r16.5

arcserve®

Arcserve Product References

This document references the following Arcserve products:

- Arcserve® Backup
- Arcserve® Unified Data Protection
- Arcserve® Unified Data Protection Agent for Windows
- Arcserve® Unified Data Protection Agent for Linux
- Arcserve® Replication and High Availability

Contact Arcserve

The Arcserve Support team offers a rich set of resources for resolving your technical issues and provides easy access to important product information.

https://www.arcserve.com/support

With Arcserve Support:

- You can get in direct touch with the same library of information that is shared internally by our Arcserve Support experts. This site provides you with access to our knowledge-base (KB) documents. From here you easily search for and find the product-related KB articles which contain field-tested solutions for many top issues and common problems.
- You can use our Live Chat link to instantly launch a real-time conversation between you and the Arcserve Support team. With Live Chat, you can get immediate answers to your concerns and questions, while still maintaining access to the product.
- You can participate in the Arcserve Global User Community to ask and answer questions, share tips and tricks, discuss best practices and participate in conversations with your peers.
- You can open a support ticket. By opening a support ticket online, you can expect a callback from one of our experts in the product area you are inquiring about.

You can access other helpful resources appropriate for your Arcserve product.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- Updated to include user feedback, enhancements, corrections, and other minor changes to help improve the usability and understanding of the product or the documentation itself.

- Updated Installation Considerations (see page 20) to include information about installing the Agent for Open Files.

- Added Restore Databases for Microsoft SQL Server 2012 AlwaysOn Availability Groups (AAG) (see page 67). This topic describes how to restore Microsoft SQL Server 2012 database when it is part of the Microsoft SQL Server 2012 AlwaysOn Availability Group.

Contents

Chapter 4: Restoring Microsoft SQL Server Databases   45

Chapter 5: Backing Up and Restoring in Cluster Environments   69

Appendix A: Troubleshooting and Disaster Recovery   81

Appendix B: Configuring Microsoft SQL Server Security Settings   97

Appendix C: Backup and Recovery Best Practices   103

Chapter 1: Introducing the Agent

This section contains the following topics:

Introduction

Arcserve Backup is a comprehensive, distributed storage solution for applications, databases, distributed servers, and file systems. It provides backup and restore capabilities for databases, business-critical applications, and network clients.

Among the agents Arcserve Backup offers is the Arcserve Backup Agent for Microsoft SQL Server. This agent enables you to perform the following actions:

- Back up your Microsoft SQL Server databases using Arcserve Backup without taking your database off-line or preventing users from adding new data

- Manage backups of Microsoft SQL Server databases remotely

- Schedule backups

- Back up to a wide array of media storage devices

- Restore Microsoft SQL Server databases using Arcserve Backup

The agent handles all communications between Arcserve Backup and Microsoft SQL Server during backup and restore jobs, including preparing, retrieving, and processing the data packets that are sent back and forth between Microsoft SQL Server and Arcserve Backup.

Architectural Overview

You can install Arcserve Backup on the same host as the Agent for Microsoft SQL Server for local operation, or on separate systems. A single installation of Arcserve Backup can work with agents on many systems, allowing multiple machines to be backed up by a single backup server. Arcserve Backup and the agent work together to back up and restore Microsoft SQL Server database objects.

The agent provides services that allow Arcserve Backup to back up and restore Microsoft SQL Server databases. The agent must reside on the same server as Microsoft SQL Server, or on a local drive of each node in a Microsoft Cluster Services environment that contains Microsoft SQL Server. In the Cluster, the agent dynamically handles the association between Microsoft SQL Server instances and Virtual Server names, and also recognizes which instances are running on which nodes.

Note: If you are using a clustered instance of Microsoft SQL Server to house your Arcserve Database, you will need to use the stand-alone installer to install the Agent for Arcserve Database on the nodes of the cluster where the instance is not active when Arcserve Backup is installed.

However, if you are installing a clustered Arcserve Database Primary Server using the Microsoft SQL Server 2008 Express Edition option for your Arcserve Database, this will not be necessary. The agent will be automatically installed on each node along with Arcserve Backup before the Arcserve Primary Server is configured for Clustered operation.

Architecturally, the agent is positioned between Arcserve Backup and Microsoft SQL Server, on the machine hosting SQL Server.

Arcserve Backup with Microsoft SQL
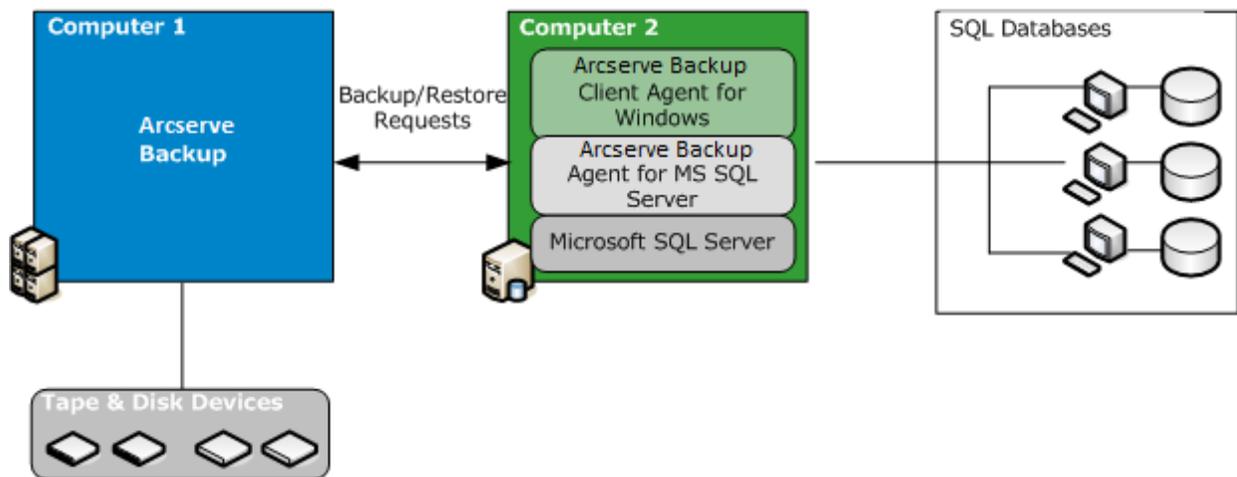
How the Agent Works

Arcserve Backup and the agent work together to back up and restore SQL Server databases. When Arcserve Backup backs up a database, it sends a request to the agent. The agent retrieves a point-in-time-consistent image of the database or its transaction log from Microsoft SQL Server in the form of a logical data stream, and sends it to Arcserve Backup, where the complete database image is backed up to media. During a restore, the agent functions in a similar fashion, and transfers the backed up database from Arcserve Backup to Microsoft SQL Server.

The agent takes advantage of the Microsoft SQL Server dump database and transaction log backup method. A dump backs up the database or transaction log in a single step. This ensures that a consistent image of the database is backed up.

For each database or transaction log submitted for backup, the agent initiates a dump in Microsoft SQL Server. Microsoft SQL Server sends the database to the agent in a series of data chunks. The agent receives the data, one chunk at a time, and passes it directly to Arcserve Backup, where it is recorded to backup media.

In a restore operation, for each database or transaction log backup being restored, the agent initiates a load operation in Microsoft SQL Server, and then returns the backed-up data to Microsoft SQL Server in the same manner that the data was provided during backup. If a single database requires more than one backup to restore it, Arcserve Backup assists in assembling the correct sequence of restore operations to be able to fully restore the database.

How a Backup Request Works

Backup jobs use the following process:

1. You start a backup job in Arcserve Backup.

2. Arcserve Backup sends the request to the agent for a database.

3. The agent retrieves a particular database or transaction log from Microsoft SQL Server, which sends multiple data chunks to the agent.

4. The agent retrieves the data chunks and transfers them to Arcserve Backup, which backs up the data to the specified storage media.

How a Restore Request Works

Restore jobs use the following process:

1. You issue a restore command from Arcserve Backup.

2. Arcserve Backup informs the agent of the restore job.

3. The agent instructs Microsoft SQL Server to prepare to receive the data.

4. Arcserve Backup accesses storage media and begins restoring data.

5. Arcserve Backup transfers data to the agent.

6. The agent transfers data to Microsoft SQL Server.

7. Microsoft SQL Server recovers the database.

How Data Flows During Backup

The following steps describe the data flow when Arcserve Backup uses the Agent for Microsoft SQL Server to back up a Microsoft SQL Server instance:

1. Arcserve Backup sends a request to the agent for a database.

2. The agent instructs Microsoft SQL Server to perform a backup of a particular database or log.

3. Microsoft SQL Server returns the data from the database in multiple chunks to the agent, one chunk at a time.

4. The agent receives the data chunks from Microsoft SQL Server and transfers them to Arcserve Backup.

5. Arcserve Backup writes the data chunks to media.

These steps are repeated until there is no more data to be backed up. The agent and the Microsoft SQL Server backup function guarantee the consistency and accuracy of the data being backed up.

Agent Services

The Agent for Microsoft SQL Server runs within the context of the Arcserve Universal Agent Service. This service is shared with other agents, providing a single point of access for backup and restore operations. The service starts automatically after the installation is completed, and dynamically recognizes the addition of new agents as they are installed.

Access Requirements

When you submit a job that includes Windows database servers, Arcserve Backup prompts you for a system user name and password for the system on which the database resides. Arcserve Backup accesses the remote servers using this user name and password.

A Microsoft SQL Server native user name and password are also required to access some database instances. When prompted by the system, enter the Microsoft SQL Server user ID and the password of the system administrator (sa), or enter a user ID and password with equivalent privileges. For more information about agent security configuration, see Configuring Microsoft SQL Server Security Settings.

Note: There are two different data transfer mechanisms available to the agent, and they have different permission requirements. A backup using Named Pipes only requires the Backup Operator permission for the specific database being backed up, and the Database Creator role to perform the restore. A backup using Virtual Devices requires the System Administrator role. However, the Named Pipes mechanism is available only for Microsoft SQL Server 7.0 and 2000. The Arcserve Backup Agent for Microsoft SQL Server requires Local Administrator permission in Windows Vista, Windows Server 2008 and later.

Backup and Restore Options

Backup options enable you to do the following:

- Perform a Full or Differential backup of a database

- Back up the entire database, a selected set of files and FileGroups within a database, or an automatically-selected set of FileGroups that contain changeable data

- Back up the transaction log of a database, with or without truncating the log

- Take the database offline automatically after a transaction log backup, leaving it in a restoring state

- Check the consistency of databases before and after backup

- Perform both a Data and Transaction Log backup of a database in a single backup job

- Include SQL Server's native error checking information as part of the backed-up data

For more information about backup options, see the section, Backing Up Microsoft SQL Server Databases (see page 29).

Restore options enable you to do the following:

■ Restore data and transaction logs

■ Automatically determine a sequence of backups to restore, to produce a live, consistent database with a single restore job

■ Use transaction log restores to restore the data to a specific point in time, or to the start or end of a named transaction

■ Restore the entire database or a selected subset of the database files and FileGroups

■ Restore a selected subset of the database as a new database

■ Perform a Torn Page Repair of a database, even while it is online

■ Leave the database in a restricted access mode

■ Preserve or clear the replication settings of the restored database

■ Change the physical location on disk of data and transaction log files

■ Check only the physical consistency of a database after restore is complete

■ Override any inconsistencies found with SQL Server's native error checking information

For more information about restore options, see the section, Restoring Microsoft SQL Server Databases (see page 45).

Microsoft SQL - Arcserve Backup Compatibility Matrix

Arcserve Backup supports features and functions introduced in various Microsoft SQL Server versions, as follows:

Feature/Function	,Microsoft SQL Server Version					
	SQL 7.0	SQL 2000	SQL 2005	SQL 2008	SQL 2012	SQL 2014
Named Pipes	Yes	Yes	No	No	No	No
Multiple Instance Support	No	Yes	Yes	Yes	Yes	Yes
Partitioning	No	No	Yes	Yes	Yes	Yes

Feature/Function	,Microsoft SQL Server Version					
	SQL 7.0	SQL 2000	SQL 2005	SQL 2008	SQL 2012	SQL 2014
Database Mirroring	No	No	Yes	Yes	Yes	Yes
FILESTREAM	No	No	No	Yes	Yes	Yes
File and FileGroup backups (see page 118)	No	Yes	Yes	Yes	Yes	Yes
Cluster Environments (see page 69)	No	Yes	Yes	Yes	Yes	Yes
Full-text search catalogs (see page 106)	No	No	Yes	Yes	Yes	Yes
Resource database	No	No	Yes	Yes	Yes	Yes
Full database backups (see page 116)	No	Yes	Yes	Yes	Yes	Yes
Differential database backups (see page 117)	No	Yes	Yes	Yes	Yes	Yes
Partial database backups (see page 120)	No	No	Yes	Yes	Yes	Yes
Torn Page Repair	No	No	Yes	Yes	Yes	Yes
Include checksum generated by SQL Server	No	No	Yes	Yes	Yes	Yes
Continue restore after checksum failure	No	No	Yes	Yes	Yes	Yes

Agent Activity Log

The Agent for Microsoft SQL Server generates a log with information about backup or restore jobs and their status. This Activity log is called sqlpagw.log and is located in the directory in which the agent has been installed. If errors appear in the Arcserve Backup job logs, check the Activity log for more information about why the errors occurred.

Online Backup of SAP R/3 Databases

When you use Microsoft SQL Server 7.0 or later as the database server for SAP R/3, you can perform an online backup of SAP R/3 databases using the Agent for Microsoft SQL Server. A separate agent for SAP R/3 is not required. The online backup procedure is the same as it is with any other database in the Microsoft SQL Server.

Note: You cannot perform offline backups of SAP R/3 databases in Microsoft SQL Server using the Agent for Microsoft SQL Server.

Agent for Arcserve Database

The Arcserve Backup Agent for Arcserve Database is a form of the Arcserve Backup Agent for Microsoft SQL Server. The agent is either installed automatically when you install Arcserve Backup, or manually using a special utility, either after the location of the Arcserve Backup database is changed, or on multiple nodes of a cluster.

This utility, named SQLAgentRmtInst.exe, is placed in the Packages sub-folder of the Arcserve Backup home directory, in a folder named ASDBSQLAgent, when you install Arcserve Backup. If you need to install the agent on a computer that is not a Arcserve Backup server, you must copy the ASDBSQLAgent folder to the system where you are installing the agent, and run the SQLAgentRmtInst.exe utility on that machine.

By itself, the Agent for Arcserve Database allows you to back up and restore the Arcserve Backup database, and the system databases and Disaster Recovery Elements from the Microsoft SQL Server instance that contains the Arcserve Backup database. When installed with the Agent for Microsoft SQL Server, it allows the Agent for Microsoft SQL Server to recognize the presence of a Arcserve Backup database, and to work with Arcserve Backup to provide the special recovery mechanisms that are available for the Arcserve Backup database.

When upgrading from a previous release of Arcserve Backup, you must upgrade the Agent for Arcserve Database. This behavior is designed to help ensure that the current version of the Arcserve Backup database is protected by the current version of the agent. As a result, you cannot clear the check box next to Agent for Microsoft SQL Server in the product selection tree on the Components dialog.
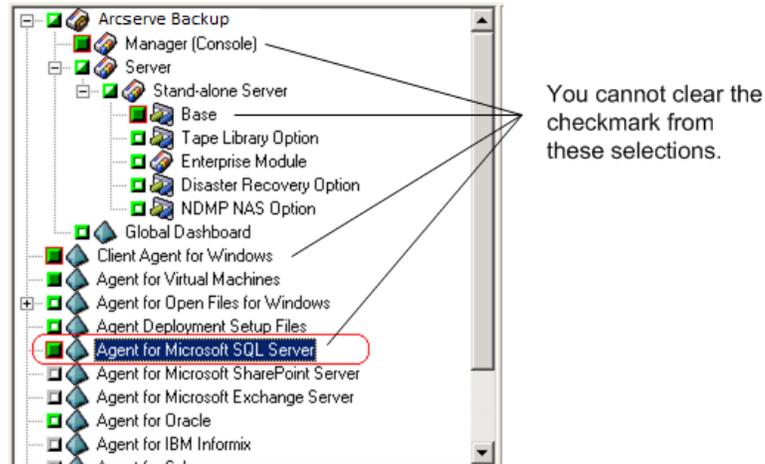
Because the Agent for Arcserve Database is a form of the Agent for Microsoft SQL Server, it will appear as the Arcserve Backup Agent for Microsoft SQL Server in the system's installed programs list. If both are present, only a single entry will appear. If you need to uninstall one or the other, the installation sequence will prompt you to select which variant to remove.

You can use the stand-alone utility that installs the Agent for Arcserve Database in any of the following situations:

■   When the Arcserve Backup database is moved

■   To re-install the agent if it is accidentally uninstalled

■   To install the agent to additional nodes of a cluster

■   To install the agent on a remote computer, if the Arcserve Backup installer is unable to do so directly

Chapter 2: Installing the Agent

The Agent for Microsoft SQL Server is a client program you can install in two configurations:

- On the same machine as Microsoft SQL Server

- On a local drive of each node in a Microsoft Cluster Services cluster that contains Microsoft SQL Server

This section explains how to install the Agent for Microsoft SQL Server in both configurations.

This section contains the following topics:

Installation Prerequisites

There is certain prerequisite information you must satisfy before installing the Agent for Microsoft SQL Server in a standard Microsoft SQL environment or a Microsoft SQL Server 2005 or later cluster environment.

For more information, see:

- Basic Prerequisites (see page 19)

- Microsoft SQL Server 2000, 2005, 2008, 2012 and 2014 Cluster Environment Prerequisites

Basic Prerequisites

Before you install the Agent for Microsoft SQL Server in a standard Microsoft SQL Server environment, review the information in the Readme to ensure you have met the basic configuration requirements.

Microsoft SQL Server 2000, 2005, 2008, 2012 and 2014 Cluster Environment Prerequisites

Before you install the Agent for Microsoft SQL Server in a Microsoft SQL Server 2000, 2005, 2008, or 2012 cluster environment, in addition to the basic prerequisites, perform the following tasks:

- Make a note of the user name and password of an MSCS domain user with system administrator privileges.

- Make a note of the Microsoft SQL Server virtual server name, cluster server user name, and cluster server password.

- Install the Agent for Microsoft SQL Server on the local drives of all nodes in the MSCS cluster as part of the initial agent installation.

Installation Considerations

The following should be considered before installing the Arcserve Backup Agent for Microsoft SQL Server:

- Installing 32-bit components on a 64-bit machine generates two copies of the file PortsConfig.cfg on the machine. The PortsConfig.cfg file for the 64-bit components is in c:\Program Files\CA\SharedComponents\CA ARCserve Backup, and the PortsConfig.cfg file for the 32-bit components is in c:\Program Files (x86)\CA\SharedComponents\CA ARCserve Backup. If you make changes to one, you must make the same changes to the other, or the components may not be able to contact one another.

- The Agent is always installed in the version native to your computer's hardware.

- The Agent requires Named Pipes and TCP/IP protocols enabled on the SQL server to communicate.

- If you uninstall a Microsoft SQL Server instance, some components used by other Microsoft SQL Server instances on the machine may be deregistered, including the Virtual Devices mechanism used to perform backups. If this occurs, the agent generates Backup Agent Error 511 ("Error creating Virtual Device Interface: COM class not found") within the error AE50015 ("Backup Failed"), or AE51004 ("Restore Failed"). To address this error, perform the following procedure:

  1. Locate the version of sqlvdi.dll that belongs to the newest version of Microsoft SQL Server on the machine.

  2. From a command prompt, change to the directory containing the selected sqlvdi.dll, and enter the following command to re-register the Virtual Devices component and re-enable backup operations:

     regsvr32 sqlvdi.dll

- Installing two or more versions of SQL Server concurrently on the same machine may create multiple versions of the SQLVDI.DLL file, which causes conflicts resulting in database backup job failures. To resolve any SQLVDI.DLL conflicts, ensure all instances of SQL Server are at the latest Service Pack, security patch or Hotfix level. You may also restart the failed SQL Server instance to reload a new SQLVDI.DLL file, or reboot the machine.

- In cluster environments, you may receive a Lost Connection error while installing the Agent for Microsoft SQL Server. This occurs in primary and stand-alone installations when you select the clustered SQL server instance as the Arcserve Backup database. To avoid this error, copy the Agent for Arcserve Backup database package to each cluster node and run setup manually. For more information, see the Arcserve Backup Administration Guide.

- To protect SQL Server instances in a cluster-aware environment, you must manually install the Agent for Microsoft SQL Server on all of the nodes in the cluster-aware environment.

- When installing the Agent for the Arcserve Backup database on other nodes in a cluster-aware environment, you must run SQLAgentRmtInst.exe, install and then run the Account Configuration Utility, DBAConfig.exe, after installation to input the correct cluster node name and password so that the Setup process can create ODBC communication.

- You do not need to install the Arcserve Backup Agent for Open Files on the Microsoft SQL Server servers that you are protecting. The Agent for Open Files is beneficial when you want to protect files that are open or in use by active applications. Because the Agent for Microsoft SQL Server is an application-specific agent that is dedicated to protecting Microsoft SQL Server servers, it offers a complete solution that encompasses all the features offered by the Agent for Open Files.

Install the Agent

Ensure that you have confirmed the installation prerequisites and performed the required pre-installation tasks. When you have completed these tasks and gathered the required information, you are ready to begin the installation process.

Note: If you have multiple versions of Microsoft SQL Server installed on the same machine, the version of SQLVDI.dll registered with the machine must be from the latest version of Microsoft SQL Server. If it is not, backup operations will fail for instances of the later versions.

For more information, see:

- Install the Agent in a Standard Microsoft SQL Server Environment (see page 22)

- Install the Agent in a Microsoft SQL Server 2000 or Later Cluster Environment (see page 22)

Install the Agent in a Standard Microsoft SQL Server Environment

To install the Agent for Microsoft SQL Server in a standard Microsoft SQL Server environment, follow the standard installation procedure for the system components, agents, and options of Arcserve Backup. For the detailed steps in this procedure, see the *Implementation Guide*.

During the installation procedure, after you select the Agent for Microsoft SQL Server for installation, the Account Configuration dialog appears.

Enter the appropriate information for each instance of your standard Microsoft Server:

- Select either SQL Server or Windows Authentication.

- Enter the user name and password of a Microsoft SQL Server user with system administrator privileges for each Microsoft SQL Server instance for which you have specified SQL Server authentication.

Install the Agent in a Microsoft SQL Server 2000 or Later Cluster Environment

To install the Agent for Microsoft SQL Server in a Microsoft SQL Server 2000, 2005, 2008, or 2012 cluster environment, follow the standard installation procedure for the system components, agents, and options of Arcserve Backup. For the detailed steps in this procedure, see the *Implementation Guide*.

During the installation procedure, after you select the Agent for Microsoft SQL Server for installation, the Account Configuration dialog appears.

Enter the appropriate cluster information for each instance of your Microsoft SQL Server 2000 or later virtual server:

- Click the cell containing the *instruction* in the Instance column to add virtual Microsoft SQL Server instances to the configuration window.

- Specify either Windows or SQL Server authentication in the Authentication column. If you specify SQL Server authentication, enter the user name and password of a Microsoft SQL Server user with system administrator (sa) rights for that instance. Confirm the password.

- Enter the name of the Microsoft SQL Server 2000, 2005, 2008, or 2012 virtual server associated with each instance.

- Enter the login ID of an MSCS domain user with system administrator privileges and the password for that user. Confirm the password.

Post-Installation Procedures

After you have installed the agent, you may need to customize the Data Transfer settings, including Virtual Device Striping parameters.

Configure Backup and Restore Parameters for the Agent for Microsoft SQL Server Using the Central Agent Admin

Use the Central Agent Admin utility to configure the Agent for Microsoft SQL Server backup and restore parameters for supported versions of Microsoft SQL Server. The parameters include settings for Microsoft Virtual Device Interface (VDI) objects and remote communication.

To configure backup and restore parameters for the Agent for Microsoft SQL Server using the Central Agent Admin

1. From the Arcserve Backup Quick Start menu, choose Administration, Central Agent Admin.

    Central Agent Admin opens.

2. In the Windows Systems tree, expand the server on which the agent is installed and then select the Agent for Microsoft SQL Server.

    Click Configuration on the toolbar.

    The Options Configuration dialog opens.

3. Click Agent for Microsoft SQL Server from the list on the left.

    The Options Configuration shows the corresponding SQL Server settings.

4. Specify the synchronized recording under Agent Log Settings as follows:

    Synchronized Recording--Forces the log messages to be written to the Activity Log as they are posted. You can disable this option to improve the performance on high-load systems by caching several messages and writing them as a group.

5. Select the Instance (ARCSERVE_DB) or the name of the instance for which you wish to change configuration for the Agent for Microsoft SQL Server.

6. Set the parameters under Virtual Device Configuration as follows:

   ■ **Number of Stripes--**Determines the number of CPUs used to perform backups. Set this value to match the number of CPUs in the database server for the fastest backup performance. The default setting is 1 and the maximum value is 32.

   ■ **Number of Buffers--**The total number of VDI buffers (of maximum transfer size) used to back up and restore. The default setting is 1. This number cannot be less than the number of stripes.

   ■ **Data Block Size (in bytes)--**All data transfer sizes are multiples of this value. Values must be a power of 2 between 512 bytes and 64 KB inclusive. The default is 65536 or 64 KB.

   ■ **Maximum transfer size--**The maximum input or output request issued by Microsoft SQL Server to the device. This is the data portion of the buffer. This parameter value must be a multiple of 64 KB. The range is from 64 KB to 4 MB. The default setting is 2097152 or 2 MB.

   ■ **Maximum VDI Wait Time - Backup (ms)--**The time, in milliseconds, a Virtual Device object waits for a response from Microsoft SQL Server during a backup operation. This setting is also used by the agent when waiting for parallel operations to synchronize or background operations to complete, including during some parts of restore operations. The default setting is 600000 ms (ten minutes).

   ■ **Maximum VDI Wait Time - Restore (ms)--**The time, in milliseconds, a Virtual Device object waits for a response from Microsoft SQL Server during a restore. Increase this time if the database to be restored contains very large data files. The default setting is 9000000 ms (2.5 hours).

7. Under Named Pipes Configuration, specify the Maximum Connection Wait Time (ms) time, in milliseconds, the Agent for Microsoft SQL Server should wait to close a named pipe if a remote connection fails. The default setting is 400 ms.

8. Set the parameters under Restore Post-Processing Wait as follows:

   ■ **Polling Period (seconds)--**The amount of time to wait between checks of the database status. The default setting is 60 seconds (one minute).

   ■ **Maximum Wait Timeout (minutes)--**The total amount of time to wait before abandoning the waiting process. If this timeout elapses and the job contains additional Transaction Log sessions to be restored, then those additional sessions may fail to restore because SQL Server is not yet ready. The default setting is 180 minutes (three hours).

   Click Apply to Multiple to display a dialog from which you can select additional SQL Servers. Click OK to apply the settings and return to Configuration.

9. Click OK to end configuration.

Configure Backup and Restore Parameters for the Agent for Microsoft SQL Server Using the Backup Agent Admin

Use the Backup Agent Admin utility to configure the Agent for Microsoft SQL Server backup and restore parameters for supported versions of Microsoft SQL Server. The parameters include settings for Microsoft Virtual Device Interface (VDI) objects and remote communication.

To configure backup and restore parameters for the Agent for Microsoft SQL Server using the Backup Agent Admin

1. Log in to the computer where the Agent for Microsoft SQL Server is installed.

   From the Windows Start menu, select All Programs, CA, ARCserve Backup, Backup Agent Admin.

   The Arcserve Backup Agent Admin dialog opens.

2. Select Agent for Microsoft SQL Server from the drop-down list and click the Configuration icon.

   The appropriate configuration dialog opens.

3. On the Common Settings tab, specify the Level of Detail and Synchronized Recording under Agent Log Settings, as follows:

   ■ **Level of Detail**--Controls the settings for level of detail of the agent's Activity Log and Debugging Log. For the Activity Log settings, a Level of Detail setting of Normal (0) includes basic information about agent activity. A setting of Detail (1) includes more detailed information about agent activity. A setting of Debug (2) enables the Debugging Log at a moderate level of detail. A setting of Trace (3) enables the Debugging Log at a very high level of detail. The Activity Log is localized for your reference. The Debugging Log is for Arcserve Support use, and is not available in multiple languages.

   ■ **Synchronized Recording**--Forces the log messages to be written to the Activity Log as they are posted. You can disable this option to improve the performance on high-load systems by caching several messages and writing them as a group.

4. Click Apply to apply the Common Settings.

5. On the Instance Settings tab, select the Instance (ARCSERVE_DB) or the name of the instance for which you wish to change configuration for the Agent for Microsoft SQL Server.

6. Set the parameters under Virtual Device Configuration as follows:

   ■ **Number of Stripes**--Determines the number of CPUs used to perform backups. Set this value to match the number of CPUs in the database server for the fastest backup performance. The default setting is 1 and the maximum value is 32.

   ■ **Number of Buffers**--The total number of VDI buffers (of maximum transfer size) used to back up and restore. The default setting is 1. This number cannot be less than the number of stripes.

   ■ **Data Block Size (in bytes)**--All data transfer sizes are multiples of this value. Values must be a power of 2 between 512 bytes and 64 KB inclusive. The default is 65536 or 64 KB.

   ■ **Maximum transfer size**--The maximum input or output request issued by Microsoft SQL Server to the device. This is the data portion of the buffer. This parameter value must be a multiple of 64 KB. The range is from 64 KB to 4 MB. The default setting is 2097152 or 2 MB.

   ■ **Maximum VDI Wait Time - Backup (ms)**--The time, in milliseconds, a Virtual Device object waits for a response from Microsoft SQL Server during a backup operation. This setting is also used by the agent when waiting for parallel operations to synchronize or background operations to complete, including during some parts of restore operations. The default setting is 60000 ms (ten minutes).

   ■ **Maximum VDI Wait Time - Restore (ms)**--The time, in milliseconds, a Virtual Device object waits for a response from Microsoft SQL Server during a restore. Increase this time if the database to be restored contains very large data files. The default setting is 9000000 ms (2.5 hours).

7. Under Named Pipes Configuration, specify the Maximum Connection Wait Time (ms) time, in milliseconds, the Agent for Microsoft SQL Server should wait to close a named pipe if a remote connection fails. The default setting is 400 ms.

8. Set the parameters under Restore Post-Processing Wait as follows:

   ■ **Polling Period (seconds)**--The amount of time to wait between checks of the database status. The default setting is 60 seconds (one minute).

   ■ **Maximum Wait Timeout (minutes)**--The total amount of time to wait before abandoning the waiting process. If this timeout elapses and the job contains additional Transaction Log sessions to be restored, then those additional sessions may fail to restore because SQL Server is not yet ready. The default setting is 180 minutes (three hours).

   Click Apply to apply the Instance Settings.

9. Click OK to end configuration.

How to Uninstall the Arcserve Backup Agent for Microsoft SQL

Uninstalling the Agent for Microsoft SQL Server is controlled through a single point from the Windows Add/Remove Programs dialog. Select Arcserve Backup and click the Remove button. The Uninstallation Application displays a list of the Arcserve Backup components installed on your system. Select the Agent you wish to uninstall and click Uninstall to automatically break the dependencies among components and uninstall them in the proper sequence.

Chapter 3: Backing Up Microsoft SQL Server Databases

This section contains information about backing up databases and Transaction logs using Arcserve Backup, the Agent for Microsoft SQL Server, and Microsoft SQL Server 7.0, Microsoft SQL Server 2000, Microsoft SQL Server 2005, Microsoft SQL Server 2008, and Microsoft SQL Server 2012.

This section contains the following topics:

Backup Overview

To *back up* is to create a copy or image of a database, Transaction Log, Database Differential, a list of changes that have been made since a previous backup, or a collection of files or FileGroups on another device (typically a media drive). Use Arcserve Backup and the Agent for Microsoft SQL Server to perform backups using the SQL Server Backup statement.

Backing up a database creates a copy of its tables, data, system objects, and user-defined objects. In the event of media failure, if you have been making regular backups of your databases and their Transaction Logs, you can recover your databases.

Important! Transaction Logs are not backed up or truncated during full or differential database backups. To back up and truncate Transaction Logs, perform a separate Transaction Log backup. When you perform the Transaction Log backup, select the Remove inactive entries from Transaction Log option to truncate the log files. For more information about Transaction Log backups see Transaction Log Backups (see page 118).

When a Microsoft SQL Server database backup is started in Arcserve Backup, the Agent for Microsoft SQL Server initiates an online backup of the database. This backup takes place while the database is active. The backup captures the state of the data at the moment the statement is executed. No partial transactions are captured. Any data changes made after the backup begins are not captured in the backed up copy of the database.

Backup Manager Agent Options

You may select backup options on a per-database basis or define a set of default options for all databases in a backup job. The Agent then applies the options to each database as appropriate.

- **Agent Options/Database Level Options--**These are agent backup options and apply to only the selected database. They can either extend or override the Global Agent options. Access Database Level Options by right-clicking the database object and selecting Agent Option from the shortcut menu.

- **Global Options/Agent Options--**These options let you specify default job options for all selected objects in the Agent type. Global Agent Options are not supported by releases of the Agent prior to r12.5. Access Global Agent Options from the Agent Options tab of the Global Options dialog.

Global Agent Options applied at a global level let you specify default job options for all databases for the Agent selected. Options selected for a specific object at the database level can either extend or override the options specified as a global option. As a general rule, options applied at the global level will extend or override options that you specify on the Job Schedule tab.

Certain options are available from only one Agent Option dialog; they are noted.

Note: The agent combines options that you specify at the database level for a specific database with the appropriate global agent options.

You can specify Global Agent Options for the Arcserve Backup components that follow:

- Agent for Microsoft SQL Server--Includes the Agent for Arcserve Backup Database, which is supported by Arcserve Backup Agent for Microsoft SQL Server, r12.5 and later.

- Agent for Virtual Machines--Supported by the Arcserve Backup Agent for Virtual Machines, r12.5 and later.

- Agent for Microsoft Exchange Server - Includes Microsoft Exchange Server Database Level and Document Level options.

Consider the behavior that follows when packaging jobs using the above-described agents:

- Global agent options are not supported by releases of the above-described agents prior to Arcserve Backup r12.5, nor are they applied if you use any other agent to back up Microsoft SQL or Exchange Server databases.

- When you upgrade an older agent to Arcserve Backup r12.5 or later, the agent applies both any pre-existing local options and all global options that apply and do not conflict with the database level (local) options.

- For jobs packaged using older agents, local options are carried over as local options.

Agent Options (Database Level)

The agent options you apply to only the selected database can either extend or override Global Option/Agent Options. These options include:

- Backup Method -- see the topic, Backup Method Options (see page 33).
  - Use Global or Rotation Options
  - Full
  - Differential

- Database Consistency Check -- see the topic, Database Consistency Check Options (see page 36).
  - Before Backup
  - After Backup
  - Continue with backup if DBCC fails
  - Do not check indexes
  - Check the physical consistency of the database

- Other options -- see the topic, Other Options (see page 38).
  - Override Global Options

Agent Options (Global Options)

Global Options/Agent Options (supported by releases of the Agent in r12.5 and later) allow you to set default job options for all SQL Server Agent objects and include essentially the same options accessible from the local options.
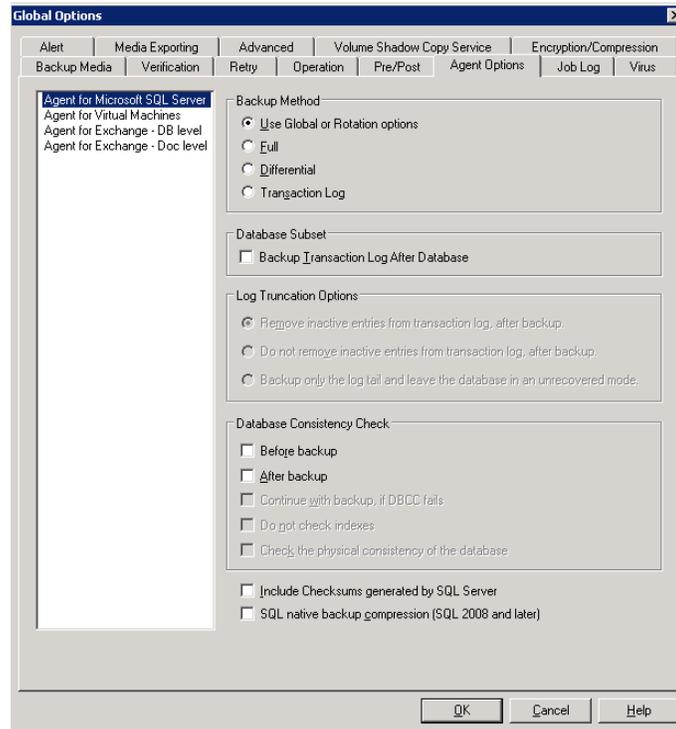
For more information:

- [Backup Method Options](#) (see page 33)

- [Database Subset](#) (see page 34)

- [Log Truncation Options](#) (see page 36)

- [Database Consistency Check Options](#) (see page 36)

- [Other Options](#) (see page 38)

Backup Method Options

The following backup methods are provided on both the Agent Options (database level) and Agent Options (Global Options) dialogs:

■ **Use Global or Rotation Options**--Use Global or Rotation Options is the default setting.

Arcserve Backup can apply Incremental and Differential global backup methods from the Job Scheduler when backing up Microsoft SQL Server databases. This lets you use rotation schemes to perform differential and Transaction Log backups of Microsoft SQL Server databases, which are dynamically adjusted based on the limitations of each individual database.

In releases of the agent that pre-date Arcserve Backup r12, the Global Backup Method or Rotation Scheme option from the Job Scheduler overrides local database options. In this release, the Global Backup Method or Rotation Scheme is applied only if you selected Use Global or Rotation Options in the database level options for the database and in the Global Agent Options for SQL Server.

This option backs up the selected database using the Backup Method from the Job Schedule. The Job Methods are applied using the logic that follows:

– The Full job method will result in a Full backup of the database.

– The Differential job method will result in a Differential backup of the database, unless this database has not yet had a Full backup.

– The Incremental job method will result in a Transaction Log backup With Truncation for databases using the Full and Bulk-Logged Recovery Models, and a Differential backup of databases using the Simple Recovery Model, unless this database has not yet had a Full backup.

– The three main System databases are exempt from the Job Method and from the Backup Method in the Global Agent Options; selecting this option for databases master, model, or msdb will always result in a Full backup.

■ **Full**--A Full backup is performed. All files included in the Database Subset selected will be backed up in their entirety.

■ **Differential**--Backs up only data that has changed since the last Full backup. For example, if you ran a complete backup of your database on Sunday night, you can run a differential backup on Monday night to back up only the data that changed on Monday.

Note: When selected in the Global Agent Options, this option is ignored by system databases. Databases that have not received a Database Full Backup will revert to a Full Backup.

- **Back up Transaction Log After Database**--Backs up only the Transaction log. This option is only available for databases using the Full and Bulk-Logged Recovery Models. For databases using the Simple Recovery Model, Arcserve Backup performs a Differential backup when you select Transaction Log Only from the Global Agent Options tab.

  Note: When selected in the Global Agents Options, this option is ignored by system databases. Databases that have not received a Database Full Backup will revert to a Full Backup.

The backup method selected on the Global Agent Options tab overrides the selection made in a job's global backup method or rotation phase in the Job Scheduler. If you select the backup method using the Global Agent Options tab, note the following:

- The three system databases (master, model, and msdb) are exempt from the Backup Method in the Global Agent Options tab.

- For databases that have not yet received a Database Full backup, Arcserve Backup ignores the backup method set in the Global Agent Options tab and performs a full backup by default.

- For databases using the Simple Recovery Model, Arcserve Backup performs a Differential backup when you choose Transaction Log Only on the Global Agent Options tab.

Because any selection other than Use Global or Rotation Method for a database overrides the selection in the Global Agent Options dialog, the Backup Method is not affected by the Override Global Options setting on the database's Agent Options (database level) dialog.

Database Subset

Database Subset options let you define the types of database components that you want to back up. You can use this option to choose between the entire database, or a selection of files and FileGroups contained within the database, when the size of the database and performance requirements do not allow you to back up the entire database.

Database Subset options are disabled if the selected Backup Method for a database is Transaction Log Only.

Important! Of the following Database Subset options, only the Back up Transaction Log After Database option is available on the Global Options/Agent Options dialog.

- **Entire Database--**Backs up the entire database.

- **Files and FileGroups--**Backs up selected files in a database. Use this option to back up a file or FileGroup when the database size and performance requirements make it impractical to perform a full database backup. This option is only available for databases using the Full and Bulk-Logged Recovery Models.

  Note: For Microsoft SQL Server 7.0 databases, Arcserve Backup performs a Files and FileGroups Full backup if you set the database subset to Files and FileGroups for the database level and backup method to Differential on the Global Agent Options tab.

- **Partial Database--**Backs up the Primary FileGroup, and any other Read-Write FileGroups. For a Read-Only database, only the Primary FileGroup will be backed up. This option requires SQL Server 2005 or later.

- **Back up Transaction Log After Database--**Backs up the Transaction Log after the database, partial database, or selected set of data files is backed up. This allows you to perform a Full backup or Differential backup and a Transaction Log backup in the same job. This option is only available for databases using the Full and Bulk-Logged Recovery Models and is ignored for databases using the Simple Recovery Model if set in the Global Agent Options dialog.

  Be aware of the following behavior:

  - If you select this option on the Global Options/Agent Options tab and specify Incremental backup method using the Job Scheduler, Arcserve Backup performs only one transaction log backup on the database and uses the Transaction Log Truncation Options from the Global Agent Options tab instead of the Incremental backup's default behavior (truncate the transaction log).

  - If you select this option using the Database Level Agent options, set the backup method at the database level to Use Global or Rotation, and set the backup method in the Global Agent Options to Transaction Log Only, Arcserve Backup performs only one transaction log backup on the database and uses the Transaction Log Truncation Options set from the database level.

  - If you select this option in the Global Agent Options dialog and specify Transaction Log Only in Database Level Options, this option and the accompanying global Log Truncation Options setting is ignored for that database.

Log Truncation Options

Log Truncation Options are accessible from the Database Level/Agent Options and Global Options/Agent Options dialogs:

- **Remove inactive entries from transaction log, after backup--**(Truncation) Truncates the Transaction Log files, removing entries included in the backup so that the space in the files can be reused. This is the default option.

- **Do not remove inactive entries from transaction log, after backup--**(No truncation) Retains backed up log entries after backup. These entries will be included in the next Transaction log backup.

- **Back up only the log tail and leave the database in unrecovered mode--**(No recovery) Backs up the log and leaves the database in a restoring state. This option is available for Microsoft SQL Server 2000 or later. Use this option to capture activity since the last backup and take the database offline prior to restoring or repairing it.

  The Log Truncation Options are available only when the selected Backup Method is Transaction Log, or when the Backup Transaction Log After Database option is checked.

  Important! Do not use the "Backup only the log tail and leave the database in unrecovered mode" log truncation option to back up the ARCserve Database. Performing a backup with this option causes the database to be placed in an offline status, and you can lose the ability to find the backups of the ARCserve Database in order to perform a restore and bring the database online. If you perform a backup of the ARCserve Database using this option, you can use ARCserve Database Recovery Wizard to recover the Arcserve Backup database and bring it back online.

Log Truncation Options are not affected by the Override Global Options setting on the Database Level Agent Options dialog. If Transaction Log or Backup Transaction Log After Database is selected for the database, the database Log Truncation Options are used.

Database Consistency Check (DBCC) Options

A database consistency check (DBCC) tests the physical and logical consistency of a database. DBCC provides the following options:

- **Before Backup--**Checks consistency before the backup of the database.

- **After Backup--**Checks consistency after the backup of the database.

- **Continue with backup, if DBCC fails--**Performs a database backup even if a consistency check before backup reports errors.

- **Do not check indexes--**Checks the database for consistency without checking indexes for user-defined tables.

  Note: The system table indexes are checked regardless of whether you select this option.

- **Check only the physical consistency of the database**--Detects torn pages and common hardware failures, but does not check the data against the rules of the database schema. It still checks the integrity of the physical structure of the page and record headers, and the consistency between the page's object ID and index ID. This option is available for Microsoft SQL Server 2000 or later. If this option is selected from the Global Agent Options tab, it is ignored for SQL Server 7.0 databases.

All error messages that are generated during the DBCC are recorded in the Agent for Microsoft SQL Server log file called sqlpagw.log. The log is located in the Backup Agent directory.

Example: How DBCC Options Work

The following example illustrates how DBCC options work in conjunction with Override Global Options on the Agent Backup Options dialog.

- With Override Global Options specified, the DBCC options selected at the database level will be the only DBCC options specified.
- With Override Global Options not specified, all of the DBCC options specified for the database and all of the DBCC options selected in the Global options will be applied together.

On the Global Options/Agent Options tab, the Database Consistency Check options that follow are specified:

- After backup
- Do not check indexes

On the Agent Backup Options dialog, Override Global Options is not selected and the Database Consistency Check options that follow are specified:

- Before backup
- Continue with backup, if DBCC fails

Note: To open the Agent Backup Options dialog, open the Backup Manager, click the Source tab, browse to and expand the Arcserve Backup server, right-click the Arcserve Backup database, and then select Agent Option from the pop-up menu.

When you submit the backup job, Arcserve Backup applies the DBCC options specified in logical order: Perform the DBCC before the backup starts. If the DBCC fails, perform the backup. After the backup is complete, do not check the indexes.

Other Options

From the Global Agent Options tab, you can specify the following additional options:

- **Include Checksum Generated by SQL Server--**Includes error checking information from Microsoft SQL Server, which can be used to validate the integrity of the backed-up data during restore. This option requires SQL Server 2005 or later and is ignored when set in the Global Agent Options dialog for SQL Server 7.0 or 2000 databases.

- **SQL Native Backup Compression**--This option applies to only SQL Server 2008 (Enterprise) and later versions. If enabled, this option directs Arcserve Backup to use SQL Server database backup compression settings, resulting in faster backup times and smaller sessions.

From the local Agent Option dialog, you can choose to Override Global Options. This setting lets you choose a backup method and database consistency check that applies to only the database selected for this job.

Dynamic and Explicit Job Packaging

Arcserve Backup provides the following two ways to package or set up your backup jobs:

- Dynamic job packaging. (see page 39)
- Explicit job packaging. (see page 40)

Dynamic Job Packaging

If you mark a database instance for dynamic job packaging when defining a backup job, Arcserve Backup automatically selects, at the time the backup job runs, all the components of the object (for example, child volumes and files) for backup.

For example, if you choose to back up an entire server and mark the server for dynamic job packaging and you change the volumes on the server, when you run the next backup job, the volumes on the server at the time of the backup are the volumes that are backed up. Any change that you make on the server marked for dynamic job packaging is included in the next backup.

Note: If you select the whole SQL Server instance for backup and then select backup options for individual databases, the individual database backup options are ignored. Only when you select the individual databases, selected per-database options are packaged and retained when you want to modify the job.

Important! When you mark a parent object for dynamic job packaging, all of its associated (or child) objects are also marked for dynamic job packaging and are selected for backup. Child objects of an object marked for dynamic packaging lose any separate options you have assigned them when the job is submitted.

Mark Objects for Dynamic Job Packaging

When you mark an object for dynamic job packaging when defining a backup job, Arcserve Backup automatically selects all the components of that object (child volumes, files, and so on) for backup.

To mark an object for dynamic job packaging

1. On the Source tab in the Backup Manager, expand the directory tree until the object you want to mark for dynamic job packaging is displayed.

2. Click the square next to the object. The square next to the object, and the squares next to all the children of the object, become completely green. In the following example, Microsoft SQL Server has been marked for dynamic job packaging. All of its children are also marked for dynamic job packaging.
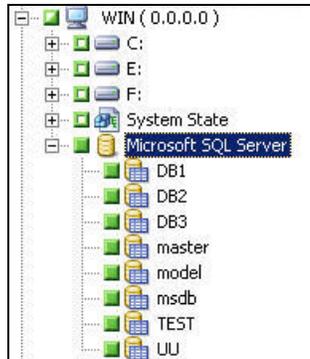
Explicit Job Packaging

To select a database object for explicit job packaging when defining your backup job, select some or all of its child objects for dynamic job packaging, but do not select the parent.

Explicit job packaging gives you the ability to customize local backup options. For example, if you run a backup job for which you have dynamically packaged the C and E drives on your server (which is explicitly packaged), you can select one set of options for drive C and another set of options for drive E.

Note: To customize volume or database options, you must package the volume or database parent items explicitly.

Mark Objects for Explicit Job Packaging

Instead of an entire database, you may wish to select a specific child object for backup, but not its parent. To customize volume and database options, mark objects for explicit job packaging.

To mark an object for explicit job packaging

1. On the Source tab in the Backup Manager, expand the directory tree until the object you want to mark for explicit job packaging is displayed.

2. Click the squares next to the children of the object. The squares next to the child objects become completely green and the square next to the parent object becomes half green and half white. In the following example, the C and D drives have been marked for dynamic job packaging. The computer on which they exist, WIN, has been marked for explicit job packaging.
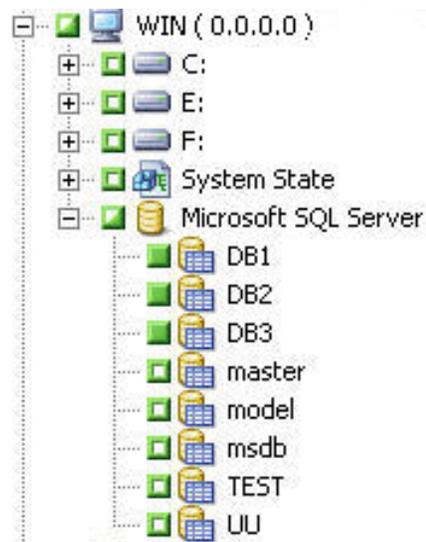
Back Up a Database

Use the Backup Manager in Arcserve Backup to back up databases.

Follow these steps:

1. Ensure that Microsoft SQL Server is running on your server. The Microsoft SQL Server service must be started.

2. Start the Arcserve Backup Universal Agent, if necessary.

   Note: This service is started automatically when the agent is installed and is set to start automatically if the machine is restarted.

3. Open the Backup Manager and locate the Microsoft SQL Server instance, listed under the name of the computer on which it is running. Expand the Microsoft SQL Server instance to display a list of databases.

4. Click Options to access Global Options and then click the Agent Options tab to access the Global Agent Options, if desired. For more information about setting Global Agent Options, see Backup Manager Agent Options. (see page 30)

5. If you are backing up from a Microsoft SQL Server 7.0 or 2000 instance, right-click the Microsoft SQL Server instance and select Transfer Method from the pop-up window. The Remote Protocol dialog opens. Select a data transfer mechanism and click OK.

   If you are backing up from a Microsoft SQL Server 2005, 2008, or 2012 instance, go to the next step.

   The default mechanism is Virtual Devices.

   Note: Select Named Pipes only if you have backup operator rights, but do not have system administrator rights, or if you cannot use Virtual Devices. Named Pipes is not available for Microsoft SQL Server 2005 and later.

   In prior releases of this agent, Virtual Devices were always used for local backups and TCP/IP backups.

6. Choose a database under the Microsoft SQL Server instance. Information about the selected database appears in the right pane of the Backup Manager.

   Note: For information about how you can select a database to apply options properly for backup, see Dynamic and Explicit Job Packaging. (see page 38)

7. Right-click the database object and select Agent Options from the pop-up window. The Agent Options dialog for Microsoft SQL Server opens. The options available vary based on the version of the Microsoft SQL Server instance and the Recovery Model of the database.

8. Select the backup method you want to perform, and the subset type you want to perform it on. For more information about backup methods and subsets, see Backup Methods.

9. If you chose the Files-and-FileGroups subset, click the Browse button. The Specify FileGroups and Files dialog opens.

   Select the specific files and FileGroups you want to back up and click OK.

10. (Optional) Enable a Database Consistency Check and select the Database Consistency Check options. For more information about Database Consistency Checks, see Database Consistency Checks and the Microsoft SQL Server documentation.

11. If you selected the Transaction Log backup type, or the Backup Transaction Log After Database option, select a Transaction Log Truncation Option.

12. Click OK.

13. Repeat the preceding steps for each database you are backing up in this job.

14. On the Destination tab in the Backup Manager, select a backup destination.

   Note: You can use the * symbol in the Group or Media fields to create partial wildcards when you select a backup destination. For example, if you have two sets of device groups, one with all members beginning GroupA and the other with all members beginning GroupB, you can select all the GroupA members by entering GroupA* in the Group field. For more information about selecting devices and media, see the *Administration Guide*.

   Click the Schedule tab and select the scheduling options for this backup. For information about scheduling backups, see the *Administration Guide*.

15. Click Submit. The Security and Agent Information dialog opens.

   Note: In this dialog, the column and button labeled Agent refer to the Client Agent for Windows, not the Agent for Microsoft SQL Server. You can edit client agent information at this time. For more information about client agents, see the *Administration Guide*.

   Verify the user name and password for the target machine and for Microsoft SQL Server. To change the security information for Microsoft SQL Server, click Security and change the information in the dialog.

16. After verifying or changing the security information, click OK. The Submit Job dialog opens.

17. (Optional) Use the Submit Job dialog to select the job execution time, submit the job on hold, add an optional job name for the backup job, or select source priority.

18. Click OK.

   The job is submitted.

   If you selected the Run Now option, the Job Status window opens. Use this window to monitor the current status of your job. For more information about the Job Status window, see the *Administration Guide*.

Chapter 4: Restoring Microsoft SQL Server Databases

This section contains information about restoring databases and Transaction logs using Arcserve Backup, the Agent for Microsoft SQL Server, and Microsoft SQL Server 7.0, Microsoft SQL Server 2000, Microsoft SQL Server 2005, Microsoft SQL Server 2008, and Microsoft SQL Server 2012.

This section contains the following topics:

Restore Options

The Agent for Microsoft SQL Server provides restore options.

- **Automatic Selection** (see page 47)

- **Restore Type** (see page 47): Database, Files or FileGroups, Partial (Microsoft SQL Server 2000 and later), or Torn Page Repair (Microsoft SQL Server 2005)

- **Log Point in Time Restore** (see page 49)

- **Recovery Completion State** (see page 50)

- **Database Consistency Check** (see page 36)

- **Force Restore to use Named Pipes** (see page 52)

- **Continue Restore After Checksum Failure** (see page 52)

- **Miscellaneous Options** (see page 52)

Automatic Selection Option

The Automatic Selection option automatically:

- Selects other sessions that must be restored with the session you are restoring for the restore job to be successful

- Applies selected options appropriately to the automatically selected sessions

The Automatic Selection option is enabled by default for every restore job. Using Automatic Selection saves you time and prevents errors in packaging restore jobs.

Important! Using Automatic Selection, you may be unable to restore a database to a different location on a disk (for example, to a different drive letter or directory path, or with a different file name) using backups from previous versions of Arcserve Backup or BrightStor® Enterprise Backup. For more information about restoring to a different location, see Restore to Alternative Disk Locations Using Automatic Selection.

Restore Types

The Agent for Microsoft SQL Server supports the following types of restores:

Database restore

Restores the entire database. If the selected session is a Database Differential backup or a Partial Full backup, the last prior Database Full backup is required as a pre-requisite. If the selected session is a Partial Differential backup, the last Database Full or Partial Full backup is required as a pre-requisite.

Transaction Log restore

Restores the Transaction log. Restoring a Transaction log is also referred to as "applying" a Transaction log. When you restore a Transaction log, Microsoft SQL Server re-executes the changes contained in the log and rolls back any transactions that were uncommitted when you backed up the Transaction log.

After restoring a database from a Full Backup, you can load the differential backup (if any) and the Transaction log backups you created after backing up that database. Loading Transaction logs lets you to recover as much of a database as possible.

You must load backups of the Transaction log in the sequence in which they were created. Microsoft SQL Server checks the timestamps on each backed up database and each backed up Transaction log to verify that the sequence is correct.

After Microsoft SQL Server has loaded the entire sequence of Transaction log backups, the database is restored to its state at the time of the last Transaction log backup, excluding any uncommitted transactions. The only uncommitted transaction that Microsoft SQL Server does not roll back is the Microsoft SQL Backup Log transaction, which is completed as part of the restore process instead.

Note: A Transaction Log restore must be applied to the original database or to a restored copy of the original database. Otherwise, the logs will not be applicable.

Files and FileGroups restore

Restores selected files and FileGroups. You can restore files and FileGroups from either a Files-and-FileGroups backup, Partial backup or Database backup. When restoring files or FileGroups, you must first restore from a Full backup session, then optionally from a differential session, followed by all of the Transaction log backup sessions performed after the Full or Differential backup. When you restore a file or FileGroup, you must apply the Transaction log to the database files immediately after the last file or FileGroup operation.

Note: A Files and FileGroups restore must be applied to the original database or to a restored copy of the original database. Otherwise, the logs will not be applicable.

Partial restore

A partial restore always restores the primary FileGroup and any other FileGroups you specify as a new database. The result is a subset of the database. FileGroups that are not restored are marked as offline and are not accessible.

Note: A database that is created using a Partial Restore may not be eligible for a Database backup, due to the unrestored FileGroups being offline. To resolve this condition, either restore the remaining FileGroups to the partially-restored database, or remove them from the database structure using the Microsoft SQL Server Enterprise Manager or Management Studio.

Torn Page Repair

Restores only the pages of data on disk which have been marked as damaged. SQL Server 2005 has the ability to isolate the damaged parts of a database while leaving the rest intact. When this happens with a Full Recovery Model database, the database may be eligible for a Torn Page Repair, which is much faster than a Database restore. The dependencies for this restore are similar to a Files and FileGroups restore.

There are both Online and Offline versions of this operation. In an Online Torn Page Repair, the database remains online the entire time, and undamaged tables are still accessible. In an Offline Torn Page Repair, the database should be taken offline using a Log Tail backup before the restore is performed. Online Repair requires the Enterprise Edition of SQL Server.

Note: Torn Page Repair restore must be applied to the original database, or to a restored copy of the original database. Otherwise, the data and logs will not be applicable.

Important! When using Automatic Selection, you may be unable to restore a database to a different location on a disk (for example, to a different drive letter or directory path, or with a different file name) using backups from previous versions of Arcserve Backup or BrightStor® Enterprise Backup. For more information about restoring to a different location, see Restore to Alternative Disk Locations Using Automatic Selection.

Log Point-in-Time Restore Options

The Log Point-in-Time Restore option restores a database to the state it was in at a date and time, or as of a named transaction, that you have specified. You should use Automatic Selection with this option. This option is only available when the selected session is a Transaction Log backup.

Important! You cannot use the Log Point in Time Restore option if the database you are recovering uses the Bulk-Logged recovery model.

To find the correct log when you have selected the Log Point in Time Restore option, Microsoft SQL Server restores the record in each Transaction log backup containing the start and finish time of the backup. Microsoft SQL Server then searches this record for the time you have specified.

- If Microsoft SQL Server finds the specified time, it restores the log to the point in the record that contains the time you submitted. The agent then signals Arcserve Backup to stop restoring and the database is fully recovered. If there are other logs with the same time, those logs are ignored and the subsequent sessions are skipped.

- If the specified time occurs after those contained in the log, Microsoft SQL Server restores the log and leaves the database in a restoring state, waiting for the next log restore operation.

- If the specified time occurs before those contained in the log, Microsoft SQL Server cannot restore the log.

The Log Point in Time Restore option has limitations. For example, if you do not select Automatic Selection and choose the Force Restore Over Existing Files option and you restore one or more logs belonging to the same database, but you do not select the appropriate database, differential backup, and FileGroup sessions to be restored first, the job will be incomplete and the subsequent sessions for that database will be ignored.

The options available for Log Point in Time Restore are:

Stop at time

Includes date and time fields in which you can enter a specific date and time. The option recovers the database to the specified date and time. This is the default option.

Stop at log mark

The option recovers the database activity through the transaction which is marked with the specified name, including the transaction that contains the mark. If you do not select the After Date and Time option, recovery stops at the first marked transaction with the specified name. If you select the After Date and Time option, recovery stops at the first marked transaction with the specified name exactly at or after the specified date and time.

Note: This option is available in Microsoft SQL Server 2000 or later.

Stop before log mark

The option recovers the database activity prior to the transaction which is marked with the specified name. The transaction which contains the mark is not re-executed. If you do not select the After Date and Time option, recovery stops at the first marked transaction with the specified name. If you select the After Date and Time option, recovery stops at the first marked transaction with the specified name exactly at or after the specified date and time.

Note: This option is available in Microsoft SQL Server 2000 or later.

After Date and Time

Allows you to specify a point in time after which Microsoft SQL Server will look for the specified Log Mark. The recovery stops at the specified mark only when the timestamp for the log mark is later than the specified time. The same Date and Time entry fields are used as for the Stop at Time option. Use this option with the Stop at Log Mark or Stop Before Log Mark options.

Recovery Completion State Options

The Recovery Completion State options let you specify the final state of a database after restore. The following options are available:

Leave database operational. No additional transaction logs can be restored

Instructs the restore operation to roll back any uncommitted transactions. After the recovery process, the database is ready for use.

Note: If you use Automatic Selection, you do not have to choose any of the Recovery Completion State selections manually for each session. Arcserve Backup performs the selection of sessions and applies the appropriate options automatically to each session. If you do not choose Automatic Selection, you must follow Microsoft SQL Server rules regarding the restore flow.

For more information, see Microsoft SQL Server documentation.

Leave database nonoperational, but able to restore additional transaction logs

Instructs the restore operation *not* to roll back any uncommitted transactions and to leave the database in a state where it can accept additional Files-and-FileGroups, Differential, or Transaction Log restores. You must choose either this option or the Leave database read-only option to apply another differential backup or Transaction log after this restore job is complete. This option is generally used when restoring a database from multiple sessions without using Automatic Selection.

Leave database read-only and able to restore additional transaction logs

Prepares a standby (warm backup) database. A standby database is a second database on a different server which you can bring online if the primary production server fails. It contains a copy of the database on the primary server. The database is brought online in a special form of Read-Only state which can be revered to a Restoring state if another restore occurs. An Undo File is created during the restore which contains the information SQL Server needs to make this transition. The location and filename of the Undo File must be specified in the restore options.

Note: Standby databases are not eligible for backup. If a standby database is explicitly selected for backup, the backup will fail. If a SQL Server instance which contains a standby database is selected for backup, the standby database will be excluded by the agent.

For more information about standby servers, see Microsoft SQL Server documentation.

Database Consistency Check (DBCC) Options

A database consistency check (DBCC) tests the physical and logical consistency of a database. DBCC provides the following options:

After restore

Performs DBCC after the restore of the database.

Before restore

Performs DBCC before an Online Torn Page Repair restore of the database. (SQL Server 2000 or later, Enterprise Edition only.)

Do not check indexes

Checks the database for consistency without checking indexes for user-defined tables.

Note: The system table indexes are checked regardless of whether you select this option.

Check only the physical consistency of the database

Detects torn pages and common hardware failures, but does not check the data against the rules of the database schema. It still checks the integrity of the physical structure of the page and record headers, and the consistency between the page's object ID and index ID. This option is available for Microsoft SQL Server 2000 or later. If this option is selected from the Global Agent Options tab, it is ignored for SQL Server 7.0 databases.

All error messages that are generated during the DBCC are recorded in the Agent for Microsoft SQL Server log file called sqlpagw.log. The log is located in the Backup Agent directory.

Force Restore to use Named Pipes

This option forces the agent to use Named Pipes to return the data to the Microsoft SQL Server. This provides an alternate mechanism for restoring a session if the Virtual Device Interface is not functioning properly. This option is available for Microsoft SQL Server 7.0.

Continue Restore After Checksum Failure

This option allows Microsoft SQL Server 2005 to continue processing a restore if it detects an inconsistency between the data and checksums included in the backup.

Miscellaneous Options

The following are the miscellaneous options you can select:

- Force Restore Over Existing Files (see page 53)
- Restricted User Access After Restore Option (see page 53)
- Keep Replication Settings Option (see page 53)
- Use Current Arcserve Database as Original Location (see page 53)
- Preserve Current Arcserve Domain Memberships (see page 53)

Force Restore Over Existing Files and Databases

Lets Microsoft SQL Server overwrite files it does not recognize as part of the database it is restoring. Use this option only if you receive a message from Microsoft SQL Server prompting you to use the With Replace option.

Microsoft SQL Server supports this option for database restore and file or FileGroup restore operations.

Important! Microsoft SQL Server 2005 will refuse by default to overwrite an online database using the Full or Bulk-Logged Recovery Model. Instead, it produces an error message saying to either take the database offline by performing a Log Tail backup, or restore with the "WITH REPLACE" option. Selecting this option applies the "WITH REPLACE" option to the restore, and forces SQL Server to overwrite the existing database.

Restricted User Access After Restore Option

Restricts access to a newly restored database to members of the db_owner, dbcreator, or sysadmin roles. In Microsoft SQL Server 2000 and Microsoft SQL Server 2005, Restricted_User replaces the DBO_Only option from Microsoft SQL Server 7.0. This option requires the Leave database operational, no additional transaction logs can be restored option.

Keep Replication Settings

Instructs the restore operation to preserve replication settings when restoring a published database to a server other than the one on which it was created. This prevents Microsoft SQL Server from resetting the replication settings when it restores a database or log backup on a warm standby server and recovers the database. Use the Keep Replication Settings option when setting up replication to work with log shipping.

You cannot select this option when restoring a backup with the Leave database non-operational, but able to restore additional transaction logs option. Use this option only with the Leave database operational, no additional transaction logs can be restored option.

Use Current Arcserve Database as Original Location

Overwrites the current Arcserve Database instead of the database which was backed up to this session when a restore to Original Location is selected. You can use this option to migrate the session and log information from one Arcserve Domain to another.

Preserve Current Arcserve Domain Memberships

Retrieves the current information about Arcserve Domains, such as the Arcserve Domain name, Primary Server identity, and Member Server identities from the destination database before the restore begins. This information is written back after the restore is completed so that the information is preserved even after the restore. This option is enabled when the Automatic Selection, Leave Database Operational and Use Current Arcserve Database as Original Location options are selected.

Database File Options

Using the Database File options, you can:

- View the list of files for the database and related information.

- Select the files to be restored in a Files and FileGroup restore, or the FileGroups to be restored in a Partial Restore.

- Change the location or file names of files during a restore. Use the location change function only when restoring from Full backups, or when using Automatic Selection.

You can move the files to a different location on disk either individually or using Move Rules. Move Rules can be applied to an entire database, individual FileGroup or Transaction Log, or to an individual file. You can specify rules to move files to a particular drive, directory path, or rename the file, all independently of each other. If you want to rename files at the Database or FileGroup level, you can specify a filename change using a wildcard pattern.

To change database file locations

1. On the Agent Restore Options dialog, select the Database File Options tab.
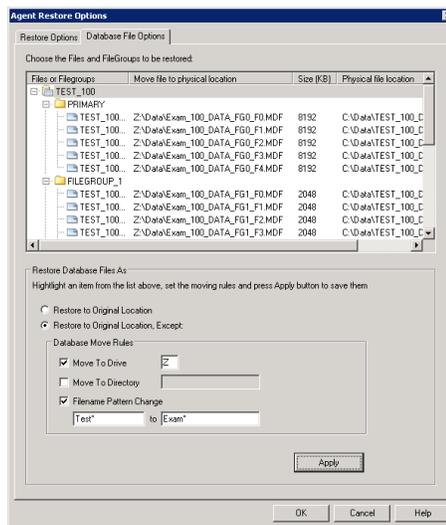
2. In the navigation tree, select one of the following options to apply the change:

- Select the database, if you want to apply the rule to all of the files in the database.

- Select a FileGroup or the Transaction Log, if you want to apply the rule to the files in a particular FileGroup or the Transaction Log.

- Select the file, if you want to change just a single file.

3. Under Restore Database Files As, select one of the following options, as appropriate:

    Restore to Original Location

    Available at the Database level. Clears any changes to the drive letters, paths, and file names. You must still click the Apply button after selecting this option for the change to take effect.

    Restore to Original Location, Except

    Available at the Database, FileGroup and Transaction Log, and File levels. Applies the requested changes to the drive letter, paths and file names based on the location of the file when the backup was performed.

    Inherit Move Rules, Except

    Available at the FileGroup and Transaction Log, and File levels. Applies the requested changes to the drive letter, paths and file names based on the changes which have already been made.

4. In the box labeled Database Move Rules, FileGroup Move Rules, or File Move Rules, select one or more of the following options:

    ■   Select the Move To Drive check box and enter a different drive letter in the field beside it.

    ■   Select the Move To Directory check box and enter a different directory path in the field beside it.

    ■   Select the Filename Pattern Change check box, to change the filenames for the entire database, FileGroup, or Transaction Log. Enter a wildcard pattern that matches the names of the files you want to rename in the field below and enter the wildcard pattern that you want it to be renamed to in the to field.

        For example, if you want to rename all the files that begin with Group as Members, enter Group* in the field and Member* in the to field.

    ■   Select the Rename File check box and enter a different file name, to rename a single file.

5. Click the Apply button.

   The changes are applied to the tree.

   Note: If you make a different selection or close the Agent Restore Options dialog without applying the changes, the selections you made will be lost.

   If the rules you specify result in two files being given the same physical location on disk, a red indicator appears at the bottom of the dialog, and in the tree next to both the affected files and the object where the rule was applied.

6. Repeat steps 2 through 5 for each change you need to make.

   Note: If you are using a wildcard pattern to rename files, and the pattern for the original filenames does not match with one or more of the files to which it would be applied, a yellow indicator will appear at the bottom of the dialog, and in the tree next to both the affected files and the object where the rule was applied.

Restore Databases Using Restore by Tree Method

To restore using the Restore by Tree method

1. On the Restore Manager Source tab, select Restore by Tree from the drop-down list.

2. In the navigation tree, expand the computer from which the database was backed up to view the database instances. Click the yellow database icon to expand the database instance containing the database you want to restore and click the database name to select it.

3. To use the most recent backup, go to the following step.

   To use a backup other than the most recent one, click the first Recovery Point drop-down control to select a recovery day. Next, click the second Recovery Point drop-down control to select a recovery session.

4. Right-click the database name you selected and choose Agent Options from the pop-up window. The Agent Restore Options dialog appears. The dialog varies depending on the backup method of the selected session and the version of SQL Server from which the database was backed up.

5. Perform one of the following to select restore options:

   ■ Click OK to accept the default options and allow the Automatic Selection option select the appropriate restore sequence and options for the restore job. The Automatic Selection option is the default option for every restore job.

   ■ Manually select the options you want to use for this restore sequence and click OK. For more information about options, see Restore Options.

   Important! The Automatic Selection option on the Agent Restore Options dialog automatically selects the sessions which must be restored and applies the selected options appropriately to each session.

6. On the Source tab, ensure that the session you want to restore is selected.

7. In the Restore Manager, click the Destination tab and select a destination using one of the following procedures:

   - To restore to the original instance on the original server using the original database name, select the Restore files to their original locations option if not selected already.

   - To restore to a different server but to an instance of Microsoft SQL Server with the same version and instance name, clear the Original Location check box and select the destination machine. The destination machine must have an instance with the same name as the original and the same version of Microsoft SQL Server or higher.

   - To restore to a different server or a different instance on the original server using the original database name, clear the Restore files to their original locations option and select the Microsoft SQL Server instance for the destination server.

   - To restore using a different database name, clear the Restore files to their original locations option, select the destination server, and select the Microsoft SQL Server instance on the destination server. Enter a backslash and the new name of the database at the end of the displayed path, as in the following examples:

     \\SERVER1\MSSQLSERVER\Lightning

     \\SERVER2\WEATHER\Thunder

     Note: If you are restoring a database backed up from a Microsoft SQL Server 7.0 instance to a Microsoft SQL Server 2000, Microsoft SQL Server 2005, 2008, 2012 or 2014 instance, you must explicitly select the target instance even if it is a default instance.

8. Click Submit.

   The Restore Media dialog appears.

9. Select the backup server where the restore job is going to run and click OK.

   The Session User Name and Password dialog opens.

10. Verify or change the user name or password for the Windows computer on which Microsoft SQL Server is loaded. To verify or change the user name or password, follow these steps:

    a. Select a session on the Machine tab and click Edit. The Enter User Name and Password dialog opens.

    b. Enter or modify the user name and password.

    c. If a session password was assigned to this session, enter the session password.

    d. To apply the user name and password you entered to all the sessions you are restoring, select the Apply [User Name and Password] to all rows option.

    e. Click OK.

11. Verify or change the user name or password for the database servers to which you are restoring. To verify or change the user name or password for the database servers, follow these steps:

    a. Select the DBAgent tab.

    b. Select a session and click Edit. The Enter User Name and Password dialog opens.

    c. Enter or modify the user name and password.

    d. If you want the user name and password you entered applied to all the sessions you are restoring, select the Apply [User Name and Password] to all rows option.

    e. Click OK.

12. Click OK on the Session User Name and Password dialog. The Submit Job dialog opens.

13. (Optional) Use the Submit Job dialog to select the job execution time, submit the job on hold, add an optional job name for the backup job, or select source priority.

14. Click OK to submit this job. If you selected Run Now, the Job Status window opens. Use this window to monitor your job. For more information about the Job Status window, see the *Administration Guide*.

Restore Databases Using Restore by Session Method

To perform a restore operation using Restore by Session method

1. On the Restore Manager Source tab, select Restore by Session from the drop-down list. A list of the media you have used when backing up with Arcserve Backup opens.

2. To create a filter to view only the sessions from a specific server or the sessions from a specific database on a specific server, perform the following procedure:

    a. Select the Filter tab. The Filter dialog opens.

    b. Click the SQL Agent Filter tab. The SQL Agent Filter dialog opens.

    c. Enter a machine name to restore the sessions from a particular server, or a machine name and database name to restore the sessions of a particular database.

       If you are using Microsoft SQL Server 2000 or SQL server 2005, you can also enter a machine name, instance name, and database name to restore a database from a specific instance of Microsoft SQL Server.

    d. Click OK.

    Note: Once you apply the filter settings, expand the media item to view the results. If the media item was already expanded, collapse it and expand it again to see the results.

3. Expand the media containing the backup you want to restore and select the session containing the specific database or log you want to restore.

    Note: Microsoft SQL Server backups have one Database, Partial, Files-and-FileGroups, or Transaction log backup for each session on the media.

4. Right-click the session containing the backup session you want to restore and choose Agent Options from the pop-up window.

    The Agent Restore Options dialog appears. This dialog varies depending on the backup method of the session selected, and the version of SQL Server from which the database was backed up.

5. Perform one of the following to select restore options:

    ■ Click OK to accept the default options and allow the Automatic Selection option select the appropriate restore sequence and options for the restore job. The Automatic Selection option is the default option for every restore job.

    ■ Manually select the options you want to use for this restore sequence and click OK. For more information about options, see Restore Options.

       Important! The Automatic Selection option on the Agent Restore Options dialog automatically selects the sessions which must be restored and applies the selected options appropriately to each session.

6. On the Source tab, ensure that the session you want to restore is selected.

7. In the Restore Manager, click the Destination tab and select a destination using one of the following procedures:

   ■ To restore to the original instance on the original server using the original database name, select the Restore files to their original locations option if not selected already.

   ■ To restore to a different server but to an instance of Microsoft SQL Server with the same version and instance name, clear the Original Location check box and select the destination machine. The destination machine must have an instance with the same name as the original and the same version of Microsoft SQL Server or higher.

   ■ To restore to a different server or a different instance on the original server using the original database name, clear the Restore files to their original locations option and select the Microsoft SQL Server instance for the destination server.

   ■ To restore using a different database name, clear the Restore files to their original locations option, select the destination server, and select the Microsoft SQL Server instance on the destination server. Enter a backslash and the new name of the database at the end of the displayed path, as in the following examples:

     \\SERVER1\MSSQLSERVER\Lightning

     \\SERVER2\WEATHER\Thunder

   ■ **Note:** If you are restoring a database backed up from a Microsoft SQL Server 7.0 instance to a Microsoft SQL Server 2000, Microsoft SQL Server 2005, 2008, 2012 or 2014 instance, you must explicitly select the target instance even if it is a default instance.

8. Click Submit.

   The Restore Media dialog appears.

9. Select the backup server where the restore job is going to run and click OK.

   The Session User Name and Password dialog opens.

10. Verify or change the user name or password for the Windows computer on which Microsoft SQL Server is loaded. To verify or change the user name or password, follow these steps:

    a. Select a session on the Machine tab and click Edit. The Enter User Name and Password dialog opens.

    b. Enter or modify the user name and password.

    c. If a session password was assigned to this session, enter the session password.

    d. To apply the user name and password you entered to all the sessions you are restoring, select the Apply [User Name and Password] to all rows option.

    e. Click OK.

11. Verify or change the user name or password for the database servers to which you are restoring. To verify or change the user name or password for the database servers, follow these steps:

    a. Select the DBAgent tab.

    b. Select a session and click Edit. The Enter User Name and Password dialog opens.

    c. Enter or modify the user name and password.

    d. If you want the user name and password you entered applied to all the sessions you are restoring, select the Apply [User Name and Password] to all rows option.

    e. Click OK.

12. Click OK on the Session User Name and Password dialog. The Submit Job dialog opens.

13. (Optional) Use the Submit Job dialog to select the job execution time, submit the job on hold, add an optional job name for the backup job, or select source priority.

14. Click OK to submit this job. If you selected Run Now, the Job Status window opens. Use this window to monitor your job. For more information about the Job Status window, see the *Administration Guide*.

SQL Agent Filter Options

You can use the SQL Agent Filter option in the Restore Manager to display backup sessions of a particular database belonging to a particular server name and, for Microsoft SQL Server 2000 or later, a particular instance. This option is available when you are using the Restore by Session method.

To display backup sessions of a database belonging to a server name

1. Open the Restore Manager and select Filter from the Restore menu.

2. In the Filter dialog, select the SQL Agent Filter tab.

3. If you are using Microsoft SQL Server 7.0, enter a machine name, or a machine name and database name.

   If you are using Microsoft SQL Server 2000 or later, enter a machine name and a database name, or a machine name, instance name, and database name.

   Note: If you have several databases with the same matching character set in the name, they are all displayed.

4. Click OK.

Note: Once you apply the filter settings, expand the media item to view the results. If the media item was already expanded, collapse it and expand it again to see the results.

Restore to Alternate Disk Locations Using Automatic Selection

You can restore a database to a different location on a disk (for example, to a different drive letter or directory path, or with a different file name) while using Automatic Selection only if the Agent Restore Options dialog displays file path entries for the data files.

Note: The Agent Restore Options dialog does not have the complete file list for Transaction log and differential backups made using BrightStor ARCserve Backup Version 9.0 or earlier, or BrightStor Enterprise Backup Release 10.0.

To determine whether you can use Automatic Selection to restore a database or session to a different location

1. Right-click the database, if you are using Restore by Tree, or the last backup session for this database, if you are using Restore by Session.

   A pop-up window opens.

2. Choose Agent Options.

   The Agent Restore Options dialog opens.

3. In the second folder tab, if the FileGroups and data files appear, you can use Automatic Selection. Follow the appropriate procedures in this chapter to restore your data.

Restore to Alternate Disk Locations by Individual Session

If the FileGroups and data files are not displayed in the section called Restore database files as, you must restore the sessions individually to restore them to an alternative disk location. To restore sessions to an alternative disk location individually, use one of the following methods:

- Restore by Session Using a Single Restore Job (see page 62).
- Restore by Session Using a Separate Job for Each Session (see page 63).
- Restore by Tree Using a Separate Job for Each Session (see page 64).

Restore by Session Using a Single Restore Job

Use the Restore Manager in Arcserve Backup to restore databases by session using a single restore job.

To restore databases by session using a single restore job

1. On the Restore Manager Source tab, select Restore by Session from the drop-down list. A list of the media you have used when backing up with Arcserve Backup opens.

2. Choose the media containing the backup you want to restore, expand the session containing that backup, and select the current backup session.

3. Right-click the backup session and select Agent Options from the pop-up window. The Agent Restore Options dialog opens.

4. Clear the Automatic Selection check box and select the Leave database nonoperational, but able to restore additional transaction logs option under Recovery Completion State.

   Note: You cannot restore any additional transaction logs if this option is not selected.

5. Click OK.

6. For each additional required backup of the database, select the next most recent session, open the Agent Restore Options dialog, clear Automatic Selection, and select the Leave database nonoperational, but able to restore additional transaction logs option under Recovery Completion State. Click OK.

7. For the earliest of these backup sessions (the full backup on which the others depend), make the appropriate changes to the file paths and names.

   Important! Do not edit the file names or paths for any of the sessions except the full backup session.

8. Complete the restore job packaging, and submit the restore job. For instructions on restoring by session, see the appropriate section in this guide.

Restore by Session Using a Separate Job for Each Session

If you restore databases using a separate job for each session, you can submit each job on hold and then make each job ready individually as the previous one finishes.

To package the database restore job as separate jobs

1. On the Restore Manager Source tab, select Restore by Session from the drop-down list. A list of the media you have used when backing up with Arcserve Backup opens.

2. Choose the media containing the backup you want to restore, expand the session containing that backup, and select the most recent full database backup of the database you want to restore. This is the full backup on which the more recent backup sessions depend.

3. Right-click the backup session and select Agent Options from the pop-up window. The Agent Restore Options dialog opens.

4. Clear the Automatic Selection option and edit the file names or paths as appropriate.

5. Select the Leave database nonoperational, but able to restore additional transaction logs option under Recovery Completion State.

6.   Click OK to close the Agent Restore Options dialog and submit the restore job.

7.   Select the next backup session for the database you want to restore.

8.   Right-click the backup session and select Agent Options from the pop-up window. The Agent Restore Options dialog opens.

9.   Clear the Automatic Selection option.

10. If this is **not** the last session to be restored, select the Leave database nonoperational, but able to restore additional transaction logs option under Recovery Completion State.

     If this **is** the last session to be restored, confirm that the Leave database operational. No additional transaction logs can be restored option is selected under Recovery Completion State.

11. Click OK to close the Agent Restore Options dialog and submit the restore job. For instructions about restoring by session, see the appropriate section in this chapter.

12. Repeat the preceding steps from the point at which you close the Agent Restore Options dialog and submit the restore job until all backup sessions have been submitted for the restore.

     Note: You must clear your previous selections before selecting options for the next job.

Restore by Tree Using a Separate Job for Each Session

If you are using the Restore by Tree method, you must submit each session as a separate restore job. You may want to submit each job on hold and make each job ready individually as the previous one finishes.

To restore sessions as separate jobs using the Restore by Tree method

1.   On the Restore Manager Source tab, select Restore by Tree from the drop-down list.

2.   In the navigation tree, expand the computer from which the database you want to restore was backed up. Click the yellow database icon to expand the database instance containing the database you want to restore and select the database.

3.   Click the first Recovery Point drop-down control to select a recovery point day. Click the second Recovery Point drop-down control to view all available sessions. Find the columns labeled Method and Backup Time.

     Note: Entries appear in reverse chronological order; more recent backups appear higher on the list.

4.   Select the most recent backup with the method Database and click Select.

5.   Right-click the selected database session and select Agent Options from the pop-up menu. The Agent Restore Options dialog opens.

6. Edit the file names or paths as appropriate, and select the Leave database nonoperational, but able to restore additional transaction logs option under Recovery Completion State.

7. Click OK to close the Agent Restore Options dialog and submit this restore job. For instructions about restoring by tree, see Restore Databases Using Restore by Tree Method.

8. Click Recovery Point again and select the next backup session.

9. Open the Agent Restore Options dialog. Clear the Automatic Selection option.

10. If this is **not** the last session to be restored, select the Leave database nonoperational, but able to restore additional transaction logs option under Recovery Completion State.

    If this **is** the last session to be restored, confirm that the Leave database operational. No additional transaction logs can be restored option is selected under Recovery Completion State.

11. Click OK to close the Agent Restore Options dialog.

12. Submit the restore job. For instructions about restoring by tree, see Restore Databases Using Restore by Tree Method.

13. Repeat these steps from the point at which you close the Agent Restore Options dialog and submit the restore job until all backup sessions have been submitted for a restore.

Perform an Offline Torn Page Restore Using Microsoft SQL Server 2000, 2005, 2008, or 2012

Microsoft SQL Server 2000, 2005, 2008, and 2012 have the ability to detect when the data in the database has been damaged and isolate the damage at the data page level. You can find the current list of known damaged pages in the [suspect_pages] table of system database [msdb], at any point of time. In addition to torn page detection and isolation, SQL 2005 also introduces the ability to perform a restore in which only those data pages which are damaged are overwritten. This will allow you to bring a database which is slightly damaged back into operation quickly.

Note: Do *NOT* take the database offline before you start this procedure.

Follow these steps:

1. Change to Full Recovery Model, if the database is using the Simple Recovery Model.

2. (Optional) Perform a Database Consistency Check (DBCC CheckDB) on the database to locate any additional damaged pages beyond the one already reported. This can be done as part of step #4.

3. Disconnect all clients using the database. (Otherwise, the following step will fail).

4. Perform a Transaction Log backup with the Backup only the log tail and leave the database in an unrecovered mode (Log Tail) option. If you have not performed step 2 separately, you should also select the Database Consistency Check Before Backup option and Continue Backup if DBCC Fails option.

5. Perform an Offline Torn Page Repair Restore of the database as follows:

   a. Open the Restore Manager.

   b. On the Source tab, use the Restore By Tree view to find and select the database.

   c. Open Agent Option.

   d. Confirm that Automatic Selection is selected.

   e. Under Subset, select Torn Page Repair – Offline.

   f. Under Recovery Completion State, select Leave database operational.

   g. (Optional) You may select a Database Consistency Check after the restore.

   h. Click OK.

   i. On the Destination tab, select Restore to Original Location, if it is not selected.

   j. Submit the Restore job.

6. Change to Simple Recovery Model, if you changed the Recovery Model in step 1.

7. Resume use of the database.

Perform an Online Torn Page Restore Using Microsoft SQL Server 2000, 2005, 2008, or 2012 Enterprise, Data Center, or Developer Editions

Microsoft SQL Server 2000, 2005, 2008, and 2012 have the ability to detect when the data in the database has been damaged, and isolate the damage at the data page level. At any point in time, the current list of known damaged pages can be found in the [suspect_pages] table of system database [msdb]. In addition to torn page detection and isolation, SQL 2005 also introduces the ability to perform a Restore in which only those data pages which are damaged are overwritten. This will allow you to bring a database which is slightly damaged back into service quickly.

Note: Do *NOT* take the database offline before you start this procedure.

Follow these steps:

1. Change to Full Recovery Model, if the database is using the Simple Recovery Model.

2. (Optional) Perform a Database Consistency Check (DBCC CheckDB) on the database to locate any additional damaged pages beyond the one already reported. This can be done as part of step 4.

3. Perform an Online Torn Page Repair Restore of the database as follows:

   a. Open the Restore Manager.

   b. On the Source tab, use the Restore By Tree view to find and select the database.

   c. Open Agent Option.

   d. Confirm that Automatic Selection is selected.

   e. Under Subset, select Torn Page Repair – Online.

   f. Under Recovery Completion State, select Leave database operational.

   g. (Optional) You may select a Database Consistency Check before the restore, to identify any other damaged or corrupted pages. (Note that this is the only restore type for which a DBCC is permitted before a Restore, because DBCC requires the database to be online.)

   h. (Optional) You may select a Database Consistency Check after the restore.

   i. Click OK.

   j. On the Destination tab, select Restore to Original Location, if it is not selected.

   k. Start the Restore.

4. Attempt to query the table that contained the damaged page.

5. Perform a Transaction Log backup with the default options.

6. Perform a restore of this final Transaction Log backup **without** Automatic Selection, and with the Recovery Completion State set to Leave Database operational.

7. Change to Simple Recovery Model, if you changed the Recovery Model in step 1.

8. Resume use of the database.

Restore Databases for Microsoft SQL Server 2012 AlwaysOn Availability Groups

When restoring a Microsoft SQL Server 2012 database that is part of an AlwaysOn Availability Groups (AAG), there are some considerations that you should be aware of.

If the Microsoft SQL database is part of the Microsoft SQL 2012 AlwaysOn Availability Groups (AAG), and restoring to the original location fails, perform the following tasks:

1. Remove the database to be restored away from the Availability Groups. For more information, see http://msdn.microsoft.com/en-us/library/hh213326.aspx.

2. Use the Restore Manager to restore the database to all nodes in the AAG group.

3. Add the database back to the Availability Groups. For more information, see http://msdn.microsoft.com/en-us/library/hh213078.aspx.

Chapter 5: Backing Up and Restoring in Cluster Environments

This section contains information about backing up and restoring database and transaction logs using Arcserve Backup, the Agent for Microsoft SQL Server, and Microsoft SQL Server 2000, 2005, 2008, or 2012 in a Microsoft SQL Server cluster environment.

Note: Backing up and restoring data in a Microsoft SQL Server 7.0 cluster environment is no longer supported.

This section contains the following topics:

Using Microsoft SQL Server Writer on a Windows Server 2012 in a Cluster Environment

When using Microsoft SQL Server Writer for physical or virtual nodes on a Windows Server 2012 in a cluster environment, there are four scenarios to consider:

Physical nodes:

- When the data of a hyper-V or SQL Server obtain a non-CSV volume as the storage, the node follows the same behavior as a noncluster environment.

- When the data of a hyper-V or SQL Server obtain a CSV volume as the storage, the data becomes visible and you can back up the data given that the node owns the application data. For example, N1 (Node 1 of the cluster) installs the SQL Server and saves the database file as a CSV file. The data for the SQL Server is visible and you can back up the data from the SQL Server writer from N1.

Virtual nodes:

- When the data of a hyper-V or SQL Server obtain a CSV volume as the storage, the data becomes visible and you can back up the data given that the node owns the application data. For example, there are two physical nodes: N1 and N2. N1 is the active node and owns VM V1 where V1 is visible and can be backed up from the virtual node; however if N2 owns VM V1 then V1 is not visible and cannot be backed up from the virtual node.

- When the data of a hyper-V or SQL Server obtain a non-CSV volume as the storage, the data is not visible and cannot be backed up.

Microsoft SQL Server 2000, 2005, 2008, 2012 and 2014 Cluster Environment Prerequisites

Before you install the Agent for Microsoft SQL Server in a Microsoft SQL Server 2000, 2005, 2008, or 2012 cluster environment, in addition to the basic prerequisites, perform the following tasks:

- Make a note of the user name and password of an MSCS domain user with system administrator privileges.

- Make a note of the Microsoft SQL Server virtual server name, cluster server user name, and cluster server password.

- Install the Agent for Microsoft SQL Server on the local drives of all nodes in the MSCS cluster as part of the initial agent installation.

Microsoft SQL Server 2000, 2005, 2008, and 2012 Cluster Environment

The following sections provide the procedures to back up data in Microsoft SQL Server 2000, 2005, 2008, or 2012 cluster environments.

-
-

Select Server, Protocol, Security, and Backup Type

When performing backup jobs in a Microsoft Server cluster environment, you must first make server, protocol, security and backup type selections. Next, you must select a backup destination, set the backup schedule and submit the job.

To select a server, protocol, security, and backup type when backing up in a Microsoft SQL Virtual Server environment

1. Verify that Microsoft SQL virtual server is running in your Microsoft cluster environment.

2. Start Arcserve Backup and open the Backup Manager.

3. Click Options and then click the Agent Options tab to set Global Agent Options, if desired. For more information, see Backup Manager Agent Options (see page 30).

4. On the Source tab, expand the Microsoft SQL Server virtual server name, **not** the physical nodes or the Windows virtual server. The Microsoft SQL Server virtual server instances are located under the Microsoft SQL Server virtual server object with which they are associated.

   Note: It is recommended that you browse the clustered Microsoft SQL Server instances through their associated Microsoft SQL Server virtual server names only. Browsing from any other entry point may cause backup failures if the Windows virtual server or the Microsoft SQL Server virtual server moves to a different node of the cluster. It is recommended that you do not browse through the MSCS computer name for clustered Microsoft SQL Server 2005, 2008, and 2012 instances.

   An instance of Microsoft SQL Server 2008 Express Edition that is used as the Arcserve Database of a clustered Arcserve Backup installation should be browsed through the virtual server name associated with the clustered Arcserve Backup primary server.

5. Right-click the Microsoft SQL Server instance and select Transfer Mechanism from the pop-up window.

   The Transfer Mechanism dialog opens.

6. Select Virtual Devices and click OK.

7. Right-click the Microsoft SQL Server instance and select Security.

   The Security dialog opens.

8. Verify the security information in the Security dialog and click OK.

9. Expand the Microsoft SQL Server instance to display a list of databases, and select a database.

10. Right-click the database and select Agent Options from the pop-up window.

    The Agent Backup Options dialog opens.

11. Select the type of backup you want to perform, and any subset options you wish to use. For more information about backup types, see Backup Methods (see page 33).

12. If you chose the Files-and-FileGroups subset, click the Browse Files/FileGroups button.

    The Specify FileGroups and Files dialog opens.

13. Select the files or FileGroups you want to back up and click OK.

14. (Optional) In the Agent Backup Options dialog, enable a Database Consistency Check, select Database Consistency Check options, and click OK.

    Note: For more information about Database Consistency Checks, see Backing Up Microsoft SQL Server Databases (see page 29) and the Microsoft SQL Server documentation.

15. Repeat these steps for each database or database object you are backing up in this job.

Select Backup Destination, Schedule, and Submit the Job

After making server, protocol, security and backup type selections, you can select a backup destination, set the backup schedule and submit the job.

To select the backup destination, schedule, and to submit the job

1. On the Backup Manager Destination tab, select a backup destination.

    Note: You can use the * symbol in the Group or Media fields to create partial wildcards when you select a backup destination. For example, if you have two sets of device groups, one with all members beginning GroupA and the other with all members beginning GroupB, you can select all the GroupA members by entering GroupA* in the Group field. For more information about selecting devices and media, see the *Administration Guide*.

    Click the Schedule tab and select the scheduling options for this backup job. For information about scheduling backups, see the *Administration Guide*.

2. Click Submit.

3. In the Security and Agent Information dialog, verify the user name and password for the Windows cluster where Microsoft SQL Server is running and for the Microsoft SQL Server instance. To enter or change security information for the computer or the Microsoft SQL Server instance, select the computer or the Microsoft SQL Server instance, click Security, enter the user name and password, and click OK.

    Note: We recommend that you use the user name and password of a domain administrator for machine authentication. Domain administrators are not dependent on the computer on which the Microsoft SQL Server instance is still running. Specify a domain user with the format DomainName\UserName.

4. Click OK. The Submit Job dialog opens.

5. (Optional) Use the Submit Job dialog to select the job execution time, submit the job on hold, add an optional job name for the backup job, or select source priority.

6. Click OK to submit this job. If you selected Run Now, the Job Status window opens. Use this window to monitor your job. For more information about the Job Status window, see the *Administration Guide*.

Restore by Tree in Microsoft SQL Server 2005, 2008, or 2012 Cluster Environments

You can submit each backup session as a separate restore job if you use the Restore by Tree method.

Follow these steps:

1. On the Restore Manager Source tab, select Restore by Tree from the drop-down list.

2. In the navigation tree, expand the Microsoft SQL Server virtual server name from which the database you want to restore was backed up. Click the yellow database icon to expand the database instance and click the database you want to restore to select it.

3. To use the most recent backup, skip to the next step.

   To use a backup other than the most recent backup, choose a Recovery Point session.

4. Right-click the database name you selected and select Agent Options from the pop-up window.

   The Agent Restore Options dialog opens. The dialog varies depending upon the restore sessions selected.

5. Perform one of the following to select restore options:

   ■ Click OK to accept the default options and allow the Automatic Selection option select the appropriate restore sequence and options for the restore job. The Automatic Selection option is the default option for every restore job.

   ■ Manually select the options you want to use for this restore sequence and click OK. For more information about options, see Restore Options.

   Important! The Automatic Selection option on the Agent Restore Options dialog automatically selects the sessions which must be restored and applies the selected options appropriately to each session.

6. On the Source tab, ensure that the session you want to restore is selected.

7. In the Restore Manager, click the Destination tab and select a destination using one of the following procedures:

    ■   To restore to the original instance on the original server using the original database name, select the Restore files to their original locations option if not selected already.

    ■   To restore to a different server but to an instance of Microsoft SQL Server with the same version and instance name, clear the Original Location check box and select the destination machine. The destination machine must have an instance with the same name as the original and the same version of Microsoft SQL Server or higher.

    ■   To restore to a different server or a different instance on the original server using the original database name, clear the Restore files to their original locations option and select the Microsoft SQL Server instance for the destination server.

    ■   To restore using a different database name, clear the Restore files to their original locations option, select the destination server, and select the Microsoft SQL Server instance on the destination server. Enter a backslash and the new name of the database at the end of the displayed path, as in the following examples:

        \\SERVER1\MSSQLSERVER\Lightning

        \\SERVER2\WEATHER\Thunder

        Note: If you are restoring a database backed up from a Microsoft SQL Server 7.0 instance to a Microsoft SQL Server 2000, Microsoft SQL Server 2005, 2008, or 2012 instance, you must explicitly select the target instance even if it is a default instance.

8. Click Submit.

    The Restore Media dialog appears.

9. Select the backup server where the restore job is going to run and click OK.

    The Session User Name and Password dialog opens.

10. Verify or change the user name or password for the Windows computer on which Microsoft SQL Server is loaded. To verify or change the user name or password, follow these steps:

    a. Select a session on the Machine tab and click Edit. The Enter User Name and Password dialog opens.

    b. Enter or modify the user name and password.

    c. If a session password was assigned to this session, enter the session password.

    d. To apply the user name and password you entered to all the sessions you are restoring, select the Apply [User Name and Password] to all rows option.

    e. Click OK.

    Note: We recommend that you use the user name and password of a domain administrator for machine authentication. Domain administrators are not dependent on the computer on which the Microsoft SQL Server instance is still running. Specify a domain user with the format DomainName\UserName.

11. Verify or change the user name or password for the database servers to which you are restoring. To verify or change the user name or password for the database servers, follow these steps:

    a. Select the DBAgent tab.

    b. Select a session and click Edit. The Enter User Name and Password dialog opens.

    c. Enter or modify the user name and password.

    d. If you want the user name and password you entered applied to all the sessions you are restoring, select the Apply [User Name and Password] to all rows option**.**

    e. Click OK.

12. Click OK on the Session User Name and Password dialog.

    The Submit Job dialog opens.

13. (Optional) Use the Submit Job dialog to select the job execution time, submit the job on hold, add an optional job name for the backup job, or select source priority.

14. Click OK to submit this job. If you selected Run Now, the Job Status window opens. Use this window to monitor your job. For more information about the Job Status window, see the *Administration Guide*.

Restore by Session in Microsoft SQL Server 2000, 2005, 2008, or 2012 Environments

You can restore databases by session in a single restore job using the Restore by Session method.

Follow these steps:

1. On the Restore Manager Source tab, select Restore by Session from the drop-down list. A list of the media that you have used when backing up with Arcserve Backup opens.

2. To create a filter to restore only the sessions from a specific server or the sessions from a specific database on a specific server, perform the following steps:

   a. Click the Filter tab in the Restore Manager. The Filter dialog opens.

   b. Click the SQL Agent Filter tab. The SQL Agent Filter dialog opens.

   c. Enter a machine name to restore the sessions from a particular server, a machine name and database name to restore the sessions of a particular database, or a machine name, instance name, and database name to restore a particular instance of a database and click OK.

   Note: Once you apply the filter settings, expand the media item to view the results. If the media item was already expanded, collapse it and expand it again to see the results.

   If you do not want to create a filter, go to the next step.

3. Choose the media containing the backup you want to restore, expand the session containing that backup, and select the specific database or log you want to restore.

4. Right-click the name of the database or log you want to restore and select Agent Options from the pop-up window.

   The Agent Restore Options dialog opens.

5. Perform one of the following to select restore options:

   ■ Click OK to accept the default options and allow the Automatic Selection option select the appropriate restore sequence and options for the restore job. The Automatic Selection option is the default option for every restore job.

   ■ Manually select the options you want to use for this restore sequence and click OK. For more information about options, see Restore Options.

   Important! The Automatic Selection option on the Agent Restore Options dialog automatically selects the sessions which must be restored and applies the selected options appropriately to each session.

6. On the Source tab, ensure that the session you want to restore is selected.

7. In the Restore Manager, click the Destination tab and select a destination using one of the following procedures:

    ■   To restore to the original instance on the original server using the original database name, select the Restore files to their original locations option if not selected already.

    ■   To restore to a different server but to an instance of Microsoft SQL Server with the same version and instance name, clear the Original Location check box and select the destination machine. The destination machine must have an instance with the same name as the original and the same version of Microsoft SQL Server or higher.

    ■   To restore to a different server or a different instance on the original server using the original database name, clear the Restore files to their original locations option and select the Microsoft SQL Server instance for the destination server.

    ■   To restore using a different database name, clear the Restore files to their original locations option, select the destination server, and select the Microsoft SQL Server instance on the destination server. Enter a backslash and the new name of the database at the end of the displayed path, as in the following examples:

        \\SERVER1\MSSQLSERVER\Lightning

        \\SERVER2\WEATHER\Thunder

    ■   **Note:** If you are restoring a database backed up from a Microsoft SQL Server 7.0 instance to a Microsoft SQL Server 2000, Microsoft SQL Server 2005, 2008, or 2012 instance, you must explicitly select the target instance even if it is a default instance.

8. Click Submit.

    The Restore Media dialog appears.

9. Select the backup server where the restore job is going to run and click OK.

    The Session User Name and Password dialog opens.

10. Verify or change the user name or password for the Windows computer on which Microsoft SQL Server is loaded. To verify or change the user name or password, follow these steps:

    a. Select a session on the Machine tab and click Edit. The Enter User Name and Password dialog opens.

    b. Enter or modify the user name and password.

    c. If a session password was assigned to this session, enter the session password.

    d. To apply the user name and password you entered to all the sessions you are restoring, select the Apply [User Name and Password] to all rows option.

    e. Click OK.

    Note: We recommend that you use the user name and password of a domain administrator for machine authentication. Domain administrators are not dependent on the computer on which the Microsoft SQL Server instance is still running. Specify a domain user with the format DomainName\UserName.

11. Verify or change the user name or password for the database servers to which you are restoring. To verify or change the user name or password for the database servers, follow these steps:

    a. Select the DBAgent tab.

    b. Select a session and click Edit. The Enter User Name and Password dialog opens.

    c. Enter or modify the user name and password.

    d. If you want the user name and password you entered applied to all the sessions you are restoring, select the Apply [User Name and Password] to all rows option.

    e. Click OK.

12. Click OK on the Session User Name and Password dialog.

    The Submit Job dialog opens.

13. (Optional) Use the Submit Job dialog to select the job execution time, submit the job on hold, add an optional job name for the backup job, or select source priority.

14. Click OK to submit this job. If you selected Run Now, the Job Status window opens. Use this window to monitor your job. For more information about the Job Status window, see the *Administration Guide*.

Perform Disaster Recovery in Microsoft SQL Server 2000, 2005, 2008, or 2012 Cluster Environments

If a disaster occurs, you can recover your SQL database in a cluster environment.

Follow these steps:

1. Reinstall Microsoft SQL Server and re-create the Microsoft SQL Server virtual server.

2. Reinstall the Agent for Microsoft SQL Server, if necessary.

3. Restore the Microsoft SQL Server [master] database.

4. Restart Microsoft SQL Server in normal, multi-user mode and restore the rest of the databases, beginning with msdb, and then model, ending with the distribution database for replication, if any.

Note: For more information about recovering from a disaster, see the *Disaster Recovery Option Guide*.

Appendix A: Troubleshooting and Disaster Recovery

This appendix explains the most common messages for Arcserve Backup and the Agent for Microsoft SQL Server and provides general troubleshooting information and important information about disaster recovery.

This section contains the following topics:

General Considerations for Arcserve Backup and the Agent

The following general considerations apply to Arcserve Backup and the Agent for Microsoft SQL Server:

- Arcserve Backup does not support special characters (for example, /, \, *, <, >, or ?) in file names, FileGroup names, and database names. The Arcserve Backup Agent for Microsoft SQL does not support brackets ([ ]) in database, FileGroup, or logical file names.

- When restoring a file or FileGroup, if there are no log sessions to follow the file or FileGroup session, Arcserve Backup cannot verify if the file was modified. As a result, it cannot identify the final recovery completion state. By default, it chooses the option Leave database non-operational, but able to restore additional transaction logs. Each time you back up a file or FileGroup, ensure that you back up a log immediately afterwards.The "Backup Transaction Log After Database" option is recommended.

- The Microsoft Windows Small Business Server 2003 SharePoint database is stored in a Microsoft SQL Server Desktop Engine (MSDE) instance. The SharePoint instance does not support SQL-native authentication. You must use Windows authentication for this database instance.

- When you log into Microsoft SQL Server using Windows authentication to perform Virtual Devices-based backup and restore operations, you must have system administrator rights for Microsoft SQL Server. This is a Microsoft requirement.

- The Arcserve Backup Agent for Microsoft SQL Server does not support backing up and restoring SQL Server databases where the sum of the characters representing the instance name, the database name, the file group name, and the data file name exceeds 170 characters.

- Microsoft SQL Server 2008 introduces a data type called FILESTREAM, which stores large blocks of binary data in files on disk. FILESTREAM data is stored in logical SQL Server files that exist as directories on disk. SQL Server does not properly report the size of FILESTREAM data, and the size of this data is calculated directly by the Agent. If this data includes a large number of records, this calculation may be very time-consuming and cause delays when querying database properties during Browse and Backup.

- If an error message appears when you restore and move a database that uses a long file name, rename the database files using shorter file names and perform the restore operation on the database.

- When backing up or restoring a SQL Server database, the SendTimeOut and ReceiveTimeOut parameters define how long the Arcserve Backup Backup Server waits for a response from the agent. This prevents problems such as network errors from causing jobs to wait forever for a response that might not come. Once this time-out period expires, the waiting job fails with a network error.

However, when a large remote SQL Server database is restored, particularly when it involves creation of SQL data files, such as when you restore to a different SQL Server, the Agent for Microsoft SQL cannot reply immediately because SQL Server takes some time to create the files, and the agent must wait for this to complete before it can respond.

The default timeout value is set to 1200 seconds (20 minutes). This should be extended when you restore a large SQL Server database involving large data files. A large SQL Server database (60 GB) might require the value to be as long as 7200 seconds (120 minutes). In general, the default value (1200) can handle most database restores, as well as network errors. However, if you have a network timeout error on a large SQL Server database restore operation, you should increase the time-out value. After the restore is done, you should reset the value to 1200.

The SendTimeOut and ReceiveTimeOut keys can be found on the Arcserve Backup Server machine under:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA ARCserve Backup\Base\Task\Remote

Note: These settings apply to all agents which operate under the Universal Agent service.

- Backup with Compare and Backup with Scan Tape are not performed on agent sessions.

- The Arcserve Backup Manager and the Arcserve Backup Universal Agent must be stopped before uninstalling and reinstalling Arcserve Backup Agent for Microsoft SQL Server if they are installed on the same server as the agent.

- If a TCP/IP connection fails, or if the Port number has changed, restart the Universal Agent Service.

- To support pass-through authentication of account logon events for computers in a domain, the NetLogon service must be started.

- In a cluster environment, run the Universal Agent as a Domain Administrator, rather than as LocalSystem. This is to prevent a conflict of access privileges between the agent and Microsoft SQL Server when backing up Microsoft SQL Server databases.

Upgrade Considerations for the Agent for Microsoft SQL

The following considerations relate to upgrading the Arcserve Backup Agent for Microsoft SQL:

■ After upgrading Microsoft SQL Server (for example, from SQL Server 7.0 to SQL Server 2005) or installing additional instances, you must run the Account Configuration utility to update the agent with the appropriate changes. If you do not run the Account Configuration utility, Arcserve Backup Agent for Microsoft SQL Server will be unaware of the new or changed instances, and you may not have access to additional instances or the additional features of the updated instance. You should also perform your first backup of the upgraded or new instances immediately after running the Account Configuration utility. For information about account configuration, refer to the topic, Update the Agent Account Configuration (see page 99).

Skip or Include Database Files in Backups

Effective with Arcserve Backup r12, there are two registry keys used to include or skip certain database files during backup jobs. Use of these keys is determined by the type of database agent you are using.

SkipDSAFiles Registry Key

The following agents use the SkipDSAFiles registry key:

- Agent for Oracle, Agent for SAP R/3 (r12.1 and earlier versions)
    - *.dbf
    - Control*.*
    - Red*.log
    - Arc*.001
- Agent for Domino
    - *.nsf
    - *.ntf
    - Mail.box
- Agent for Sybase
    - Physical file of Master device
    - Physical file of non-Master device
    - Physical file of Mirror device
- Agent for Informix
    - *.000

To use the SkipDSAFiles registry key

1. When performing agent backups:

   HKEY_LOCAL_MACHINE\SOFTWARE\Computer Associates\CA ARCserve Backup\ClientAgent\Parameters

2. Set the registry key to Value Name: SkipDSAFiles

   Type: DWORD

   Value: 0 to back up and 1 to skip

BackupDBFiles Registry Key

The following agents use the BackupDBFiles registry key:

- Agent for Microsoft SQL Server

  The list of data and transaction log files that are part of online databases is retrieved from Microsoft SQL Server at the start of a file backup. This list typically includes, but not exclusively:

  - *.ldf

  - *.mdf

  - *.ndf

  Except distmdl.mdf, distmdl.ldf, mssqlsystemresource.mdf, mssqlsystemresource.ldf, which cannot be skipped. Also, if a SQL Server instance is shut down, the database files will not be skipped.

- Exchange Database Level Agent/Exchange Document Level Agent

  - *.chk

  - *.log

  - Res1.log

  - Res2.log

  - *.edb

  - *.stm

- Agent for Oracle (r12.5 or later)

  - *.dbf

  - Control*.*

  - Red*.log

  - Arc*.001

To use the BackupDBFiles registry key

1. When performing agent backups:

   HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA ARCserveBackup\ClientAgent\Parameters

2. Set the registry key to Value Name: BackupDBFiles

   Type: DWORD

   Value: 0 to skip and 1 to back up (0 is default)

Agent and Arcserve Backup Error Messages

This section explains the most common error messages for Arcserve Backup and the Agent for Microsoft SQL Server.

Backup or Restore Operation Failed

Backup or restore failed.

Reason:

There are a number of reasons for backup or restore failure.

Action:

To resolve this condition, perform the following actions:

- Determine if the backup or restore failed because an incompatible database option was set. For more information, see Valid Operations with SQL Server Database Options.

- Check the Agent for Microsoft SQL Server log file sqlpagw.log for specific errors. This log is located in the Backup Agent directory.

- See the Microsoft SQL Server manual for information on backup and restore operations.

No Icon in Browse Tree

No Microsoft SQL Server icon in the the backup source or restore destination tree.

Reason:

This can occur if the Agent for Microsoft SQL Server is not installed or the Arcserve Universal Agent service is not running or not functioning, or if no agent entry exists in either of the following place in the registry:

SOFTWARE\ComputerAssociates\CA ARCserve Backup\SQLPAAdp\Instances

A SQL Server 7.0 instance is represented by a key named dbasql170. A SQL Server 2000 or later instance is represented by a key named dbasql@*instancename*.

Note: The instance name for a default instance at SQL Server 2000 or later is called MSSQLSERVER.

Action:

To address this error, verify that the Agent for Microsoft SQL Server is installed. Restart the Arcserve Universal Agent service. Check the registry for the agent entry and reinstall the Agent for Microsoft SQL Server, if necessary.

Microsoft SQL Server Error Messages

Microsoft SQL Server errors can occur for a variety of reasons and can appear in several different Arcserve Backup error messages. If a Microsoft SQL Server error occurs, find the value shown for NativeError.

3108

"RESTORE DATABASE" must be used in single user mode when trying to restore the master database.

Reason:

An attempt was made to restore the master database without starting the database server in single-user mode.

Action:

For instructions on starting the database server in single-user mode, see Restore Master Databases.

4305 or 4326

The log in this backup set terminates at …, which is too early to apply to the database.

Reason:

An attempt was made to restore transaction logs out of order, with an older log being restored after a newer one or an attempt was made to restore a transaction log after a more recent backup.

Action:

Restore the last full or differential database backup again and reapply the transaction logs in the order in which they were backed up.

For additional information, see the *Microsoft SQL Server Books Online*.

Limitations of Microsoft SQL Server Database

Microsoft SQL Server databases have the following limitations:

- If you have named a Microsoft SQL database with a special character (for example, /, *, <>, or ?), by default, Microsoft SQL Server names the file, FileGroup, or database with the same characters. To perform a backup or restore job, rename the file, FileGroup, or database so it does not include one of these special characters. In addition, the bracket characters, ([ ]) are not supported for any object names.

- If you use the Log Point in Time restore option for Microsoft SQL Server 7.0, or the Stop at time, Stop at log mark, or Stop before log mark options for Microsoft SQL Server 2000, Microsoft SQL Server 2005, and the specified time stamp or mark is not found by the Microsoft SQL Server, the database remains in a loading state and the job result is incomplete.

Replication of Microsoft SQL Server

According to Microsoft, the replication capability of Microsoft SQL Server is not specifically designed to accomplish hot backups. See the *Microsoft SQL Server Database Administrator Guide* for more information about how to back up and restore in a replication scenario.

Configure Agent Behavior for Databases Ineligible for Backup

The Agent for Microsoft SQL Server has a pre-defined set of rules that governs how databases ineligible for backup are recognized and handled. Backup eligibility is determined by database status and other properties. Arcserve Backup posts an error or warning and marks the job as Failed or Incomplete, depending on the following factors:

- why the database is ineligible

- how the database was selected for backup (explicitly or implicitly)

The following information organized by database status is provided to help you recognize default behavior and, if needed, change the default responses by adding controller values to the Agent settings in the Windows Registry on the computer on which the Arcserve Backup Agent for Microsoft SQL Server is installed. For each database status, the default behavior is listed for each selection type, as well as the setting name that changes it. The procedure for creating and setting DWORDS follows the table.

Database Status	If Database is Selected	If Whole Instance is Selected
Standby (read-only, waiting to receive additional restores)	Message Type: Error Job Result: Failure Setting Name: Standby Explicit	Message Type: Warning Job Result: Successful Setting Name: Standby Implicit
Mirror (SQL 2005 or later) Failover Mirroring	Message Type: Warning Job Result: Incomplete Setting Name: Mirror Explicit	Message Type: n/a Job Result: Successful Setting Name: Mirror Implicit
Suspect (damaged or missing files) using Simple Recovery Model	Message Type: Error Job Result: Failure Setting Name: Suspect Explicit	Message Type: Error Job Result: Failure Setting Name: Suspect Implicit
Off-line	Message Type: Error Job Result: Failure Setting Name: Offline Explicit	Message Type: Error Job Result: Failure Setting Name: Offline Implicit
Loading	Message Type: Error Job Result: Failure Setting Name: Loading Explicit	Message Type: Error Job Result: Failure Setting Name: Loading Implicit

Database Status	If Database is Selected	If Whole Instance is Selected
Recovering (in the middle of processing restored data)	Message Type: Error Job Result: Failure Setting Name: Restoring Explicit	Message Type: Error Job Result: Failure Setting Name: Restoring Implicit
Missing (database selected for backup no longer exists)	Error Message Posted Job Result: Failure Setting Name: Restoring Explicit	Message Type: n/a Job Result: n/a Setting Name: n/a
Inaccessible (specified user cannot access database for some other reason)	Message Type: Error Job Result: Failure Setting Name: Inaccessible Explicit	Message Type: Error Job Result: Failure Setting Name: Inaccessible Implicit

Note: SQL 2005 Point-In-Time Snapshots are strictly ignored.

To change behavior:

1. In the Registry Editor, find the Registry Key "HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\CA ARCserve Backup\SQLPAAdp\Common"

2. Add a new Key within "Common" called "Responses".

3. Add new DWORD using the setting names shown in the table and set the values as follows:

   0 = Use Agent's default behavior

   1 = Warning message posted. Job result is set to Successful

   2 = Error message posted. Job result is set to Failure.

   3 = Warning message posted. Job result is set to Incomplete.

   4 = Error message posted. Job result is set to Incomplete.

   Note: Removing a value entry is the same as setting a value to 0.

Additional Database Status Considerations

Database Status -- Mirror

When the entire instance is selected, Mirror databases do not appear on the database list, and are ignored. For a Mirror database to be explicitly selected for backup, it would have been selected while it was the Principal (Active) member of the Mirroring partnership, and entered the Mirror state due to a subsequent fail-over.

Database Status -- Suspect

When a database using the Full or Bulk-Logged Recovery Model is in the Suspect state, the Agent automatically attempts to perform a Transaction Log Backup with No Truncation. If this matches the backup options selected, then the results of this backup are the only indicator. If a different backup method is selected, a Warning is posted that a Transaction Log Backup with No Truncation is being attempted instead of using the selected options.

Database Status -- Missing

Because the database list is enumerated dynamically when the entire instance is selected, the Agent for Microsoft SQL Server has no knowledge of databases that have been removed from the SQL Server instance.

Microsoft SQL Server Disaster Recovery

The Agent for Microsoft SQL Server uses the Microsoft SQL Server Backup and Restore database functions, but these functions do not actually back up the physical files that make up the database. Consequently, for a restore operation to succeed, the database instance must be online.

The Master Database

For Microsoft SQL Server to run, the master database must be set up as follows:

- A master database and a model database *must* exist.

- To have a master and a model database, you must either reinstall Microsoft SQL Server, rebuild the master database using Microsoft SQL Server setup, or restore either a Microsoft SQL Server Disaster Recovery Elements session or an offline copy of the master database from media.

- After the master and model databases exist, Microsoft SQL Server must be running to execute the Restore command.

After you restore SQL Disaster Recovery Elements, you should immediately perform a normal restore of the master, msdb, and model databases. To restore the master database, Microsoft SQL server must be running in single-user mode. For more information about restoring the master database, see Restore Master Databases. (see page 129)

Potential Restore Problems

The master database tracks all of the resources allocated to Microsoft SQL Server. If you do not perform an offline backup after you have made a major change in the Microsoft SQL Server configuration, restore problems can occur.

For example, for a Microsoft SQL Server configuration with five databases in addition to the master database, you back up the master database, drop one database (detach it from Microsoft SQL Server) and delete the files that make it up. If you do not perform an offline backup and you restore the master database backup at this point, it contains information for the dropped database. As a result, Microsoft SQL Server marks the database as suspect (inaccessible by users). You must drop the database again.

To avoid such problems, perform at least one offline backup. In addition, each time you make a major change in the Microsoft SQL Server configuration (create or drop a database or add a device), you should perform an offline backup.

Performing a Full Database backup of the master, model and msdb databases in the same job generates a Microsoft SQL Server Disaster Recovery Elements backup session. This session can be used as an off-line backup for this purpose.

Suggested Database Restore Sequence

We recommend that you restore the databases in the following order to avoid conflicts:

1. Stop all applications that may be using this instance of Microsoft SQL Server. Also stop the SQL Server Agent service.

2. Restore the [master] database.

3. Restore the [msdb] database immediately after you restore the master database.

4. Restore the [model] database immediately after you restore the msdb database.

5. Restore all other databases in normal mode.

6. Restore the distribution database for replication, if one exists.

Note: These suggestions are not requirements, but following them speeds and simplifies the disaster recovery procedure. If you restore other databases in addition to the master database before restoring msdb, Microsoft SQL server loses part of the backup and restore history for the other databases when msdb is restored. For more information about restoring the master database, see Restore Master Databases (see page 129) or see the Microsoft SQL Server documentation.

Disaster Recovery Scenario

A typical disaster recovery scenario consists of the following steps:

1. Reinstall Windows, if necessary.

2. Reinstall Arcserve Backup, if necessary.

3. Reinstall the Agent for Microsoft SQL Server and the Client Agent for Windows, if necessary. (The Client Agent is needed to restore Microsoft SQL Server Disaster Recovery Elements.)

4. Perform one of the following steps as appropriate:

   - If you have a Microsoft SQL Server Disaster Recovery Elements session, restore it.

   - If an offline backup exists, restore it.

   - If you do not have an offline backup or a Disaster Recovery Elements session, and you have the Microsoft SQL 7.0 or 2000 rebuildm.exe utility, use the utility to recreate the master and model database. For SQL 2000, 2005, 2008, and 2012, this is an ability of the SQL Server installation software. For more information, see the Microsoft documentation.

   - If an offline backup or Disaster Recovery Elements backup do not exist and you do not have the Microsoft SQL rebuildm.exe utility, reinstall the Microsoft SQL Server or MSDE-based application.

5. Restore the [master] database.

6. Restart Microsoft SQL Server in normal, multi-user mode.

7. Restore the [msdb] database.

8. Restore the model database.

9. Restore all other databases and transaction logs, except the distribution database for replication.

10. If replication is being used, restore the replication database.

Appendix B: Configuring Microsoft SQL Server Security Settings

This appendix explains how to configure Microsoft SQL Server security settings for Arcserve Backup.

This section contains the following topics:

Types of Microsoft SQL Server Authentication

Microsoft SQL Server provides two types of user authentication:

- Use the active Windows login identification

- Use separate user credentials specific to Microsoft SQL Server

Although Microsoft recommends using only Windows authentication wherever possible, in certain instances it is appropriate, or even necessary, to enable Microsoft SQL Server-based authentication. For example, you must use Microsoft SQL Server authentication for Microsoft SQL Server 7.0 when the database is running in a cluster.

Authentication Requirements

For Microsoft SQL Server authentication, you must specify a user account with system administrator access privileges. By default, Microsoft SQL Server creates an account with this level of access, called *sa*, but the Agent for Microsoft SQL Server can use any account with equivalent privileges.

For Windows authentication, any account with administrator equivalency for the machine on which the database is running typically has system administrator access privileges for the database instance.

Note: A Windows or domain administrator does not automatically have system administrator access privileges for the database if the BUILTIN\Administrators login entry in Microsoft SQL Server has been removed or does not have this role, or if a separate login entry exists for this user in Microsoft SQL Server and does not have this role.

How User Authentication Is Changed

Regardless of the authentication option you choose, you must configure the Agent and confirm that the settings in Windows are configured to match. In addition, if you change the option, you must update any existing backup jobs to reflect the change. If you are using Microsoft SQL Server 2005 or later, you must update the setting separately for each instance of SQL Server.

To change user authentication and update the settings in both Windows and Arcserve Backup so that this change is reflected, perform the following process:

1. Check and change the Microsoft SQL Server authentication method.

2. Update the Agent for Microsoft SQL Server Account Configuration.

3. Check and change the ODBC (open database connectivity) settings.

4. Update existing backup jobs in Arcserve Backup.

More information regarding the steps in this process are contained in the following sections.

Check or Change the Microsoft SQL Server Authentication Method

To check or change the Microsoft SQL Server authentication method for SQL Server 7.0 or 2000

1. On the system where Microsoft SQL Server is running, open the SQL Server Enterprise Manager.

2. In the Tree pane, expand the entries below Console Root until you reach the instance of the database server.

3. Right-click the instance and select Properties from the drop-down list. The Properties dialog opens.

4. In the Properties dialog, choose the Security tab.

5. Under Authentication, select either Microsoft SQL Server and Windows to enable Microsoft SQL server-based authentication or Windows Only to enable only Windows user names and passwords.

6. Click OK.

7. Restart the SQL Server instance for the changes to take effect.

To check or change the Microsoft SQL Server authentication method for SQL Server 2005, 2008, and 2012

1. On the system where Microsoft SQL Server is running, open the SQL Server Management Studio.

2. Connect to the instance of Microsoft SQL Server that you want to change.

3. In the Object Explorer pane, right-click the instance and select Properties from the drop-down list. The Properties dialog opens.

4. In the Properties dialog, choose the Security page.

5. Under Server Authentication, select either SQL Server and Windows Authentication Mode to enable Microsoft SQL Server-based authentication, or Windows Authentication Mode to enable only Windows users.

6. Click OK.

7. Restart the SQL Server instance for the changes to take effect.

Update the Agent Account Configuration

To update the Agent for Microsoft SQL Server Account Configuration

1. From the Start menu, start the Microsoft SQL Agent Account Configuration utility. The Account Configuration dialog opens.

2. Locate the Microsoft SQL Server instance you modified in the first part of the process. For more information, see Check or Change the Microsoft SQL Server Authentication Method.

3. Use the following criteria to select either Microsoft SQL authentication or Windows authentication:

   ▪ If you have set Microsoft SQL Server for Windows Only, select Windows authentication.

   ▪ If you have selected Mixed Mode and want to be able to back up Microsoft SQL Server using only Microsoft SQL Server native users, select Microsoft SQL authentication.

   ▪ If you have selected Mixed Mode and want to be able to back up Microsoft SQL using either Windows or Microsoft SQL Server users, select Windows authentication.

4. If you have selected Microsoft SQL authentication, enter the user name and password for a Microsoft SQL Server native user with system administrator privileges.

5. If Microsoft SQL Server is running in a cluster environment, verify that the cluster-related information is correct.

6. Click Finish to apply the changes.

Check and Change the ODBC Settings

To check and change the ODBC settings

1. On the system where Microsoft SQL Server is running, from the Start Menu, select Control Panel (if necessary) and select Administrative Tools.

2. Select Data Sources (ODBC). The ODBC Data Source Administrator dialog opens.

3. On the System DSN folder tab, select the entry for the appropriate Microsoft SQL Server instance or instances. For Microsoft SQL Server 7.0, the entry is called DBASQL7. For Microsoft SQL Server 2000 and Microsoft SQL Server 2005, the default instance is called dbasql_MSSQLSERVER. Other instances use the format dbasql_ followed by the instance name.

4. Click Configure. The Microsoft SQL Server DSN Configuration dialog opens. The name field displays the same name as the entry you selected. The Description field displays SQL Backup Agent. The Server field displays the computer on which Microsoft SQL Server is running.

5. Click Next **without changing any settings**. You are prompted to indicate whether Microsoft SQL Server should use Windows authentication or Microsoft SQL Server authentication to verify the authenticity of the login ID:

    ■   If you are using Windows Only for Microsoft SQL Server, ensure that Windows authentication is selected.

    ■   If you are using Mixed Mode for Microsoft SQL Server, select either Windows authentication to enable both Windows and Microsoft SQL Server-native user logins or Microsoft SQL Server authentication to enable only Microsoft SQL Server-native user logins.

    ■   If you are using Microsoft SQL Server authentication, ensure that the specified login ID has system administrator access. If you are uncertain, use **sa**, and consult the Microsoft SQL Server documentation for instructions on setting the password for this account in Microsoft SQL Server. Reenter the password to ensure it is correct.

6. Click Next. If you have specified Microsoft SQL Server authentication, and either the login ID or password is incorrect, an error message appears. Click OK to close the error message, reenter the login ID and password, and click Next.

7. Click Next and click Finish. The ODBC Microsoft SQL Server Setup dialog opens, summarizing the settings. Click Test Data Source.

The SQL Server ODBC Data Source Test dialog opens. After a few seconds, the dialog displays the results of a quick connection test:

■ If the Microsoft SQL Server ODBC Data Source Test dialog reports that the tests finished successfully, click OK and click OK on the ODBC Microsoft SQL Server Setup dialog. You are returned to the ODBC Data Source Administrator dialog. Click OK.

■ If the SQL Server ODBC Data Source Test dialog reports any errors or failures, this is due to an error in ODBC or the Microsoft SQL Server. See the Microsoft SQL Server documentation for troubleshooting instructions.

Update the Backup Manager

To update the Backup Manager and backup jobs

1. On the system where you have installed Arcserve Backup, start Arcserve Backup and open the Job Status Manager.

2. If you have any recurring or pending backup jobs already scheduled for this Microsoft SQL Server, select the Job Queue folder tab in the Job Status Manager.

3. Select the first job you need to update, right-click and select Modify from the pop-up window. The Backup Manager will open and load the job.

4. Select the Source tab.

5. In the left pane navigation tree, expand the server on which Microsoft SQL Server is running to show the database. If the server does not appear in the tree, see the *Implementation Guide* for instructions on how to add it.

6. Right-click the database and choose Security from the pop-up window. The Security dialog opens.

7. If you are using Windows security, specify the user name for the account you are using to access the server and either enter the password or leave the field blank.

   If you are using Microsoft SQL Server security, enter the user name and password for the Microsoft SQL Server account and click OK.

8. Click Submit to resubmit the backup job.

9. Repeat these steps for each applicable backup job.

Note: If you have any restore jobs scheduled for this Microsoft SQL Server, you must delete and recreate them.

Appendix C: Backup and Recovery Best Practices

Data loss can be limited to a file on a single disk drive or encompass your entire data center. Ideally, you should consider a strategy that protects each database you maintain. Consider database size and activity as well as business importance as you design your strategy. The decisions you make during this critical analysis phase govern the backup and restore options available in Arcserve Backup. The following information is provided to start your planning efforts.

Important! You should refer to the Microsoft SQL Server documentation for complete information on configuring SQL Server databases for optimal protection.

This section contains the following topics:

Microsoft SQL Server Database Basics

To achieve optimal results using Arcserve Backup to back up your Microsoft SQL Server databases, the following database fundamentals are provided.

A database is a collection of tables, which are comprised of rows or records. Each row is made up of columns that contain some sort of structured information about a record. Database data is stored across two operating system files -- data files and log files:

■ **Data Files**--Contain user data and metadata.

– **Primary Data File**--Contains information about the database itself including references to the other files comprising the database. In a simple database, user-defined data and objects can be stored in the Primary Data File, though Secondary Data Files should be used for such data. Each database has one primary data file (.mdf).

– **Secondary Data Files**--Contain user-defined data and objects, for example, sales figures, employee information, or product details. Use secondary data files (.ndf) to store database information across multiple disks and to manage database growth.

Data Files can include full-text search catalogs, which allow you to search on the full-text catalogs stored with a database. Though not actually data files, FILESTREAM data allows you to store unstructured data including video, photographs and other documents usually stored outside the database and are treated as files in the database.

■ **Log Files**--Contain information at the transaction level required to restore your database to a specific point in time in the event of a disaster. There is at least one log file (.ldf) per database though you may add more as needed.

Though SQL Server supports single-disk systems, you should store your Data and Log Files on separate disks.

About Files and FileGroups

All databases have a primary FileGroup. By default, the Primary Data File and any Secondary Data Files you create are stored in the Primary FileGroup. However, you may create user-defined FileGroups to store your Secondary Data Files. Use FileGroups to manage database performance, growth, and data allocation.

Note: A file can be part of only one FileGroup. A file or FileGroup can be used by only one database. Transaction log files are not contained in FileGroups at all.

Example

The following diagram illustrates a database with six user-defined Secondary Data Files (.ndf) grouped in two user-defined FileGroups and stored on three separate disks. You can create a table on each FileGroup so that any queries against the data in a table can then be allocated across the three disks in parallel, to improve performance. Note also that the Transaction Log and Primary Data Files are stored separately from user data.
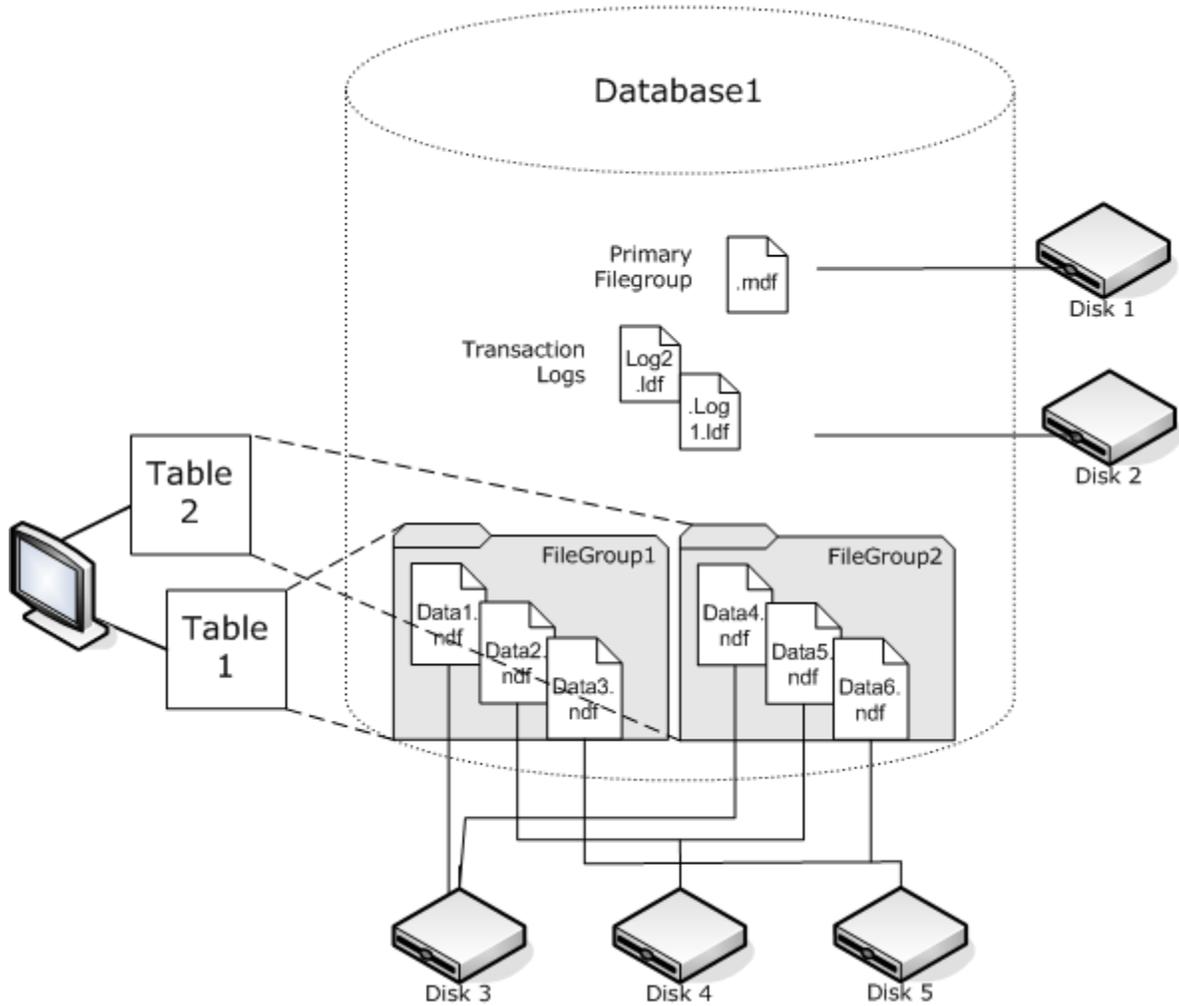
Full-Text Search Catalogs

With SQL Server 2005, Microsoft added full-text search enhancements that let you back up and restore one or more full-text catalogs. You can now backup catalogs with, or separate from, your database data.

A database can hold one or many full-text catalogs, but a catalog can only belong to one database. A full-text catalog holds the full-text indexes created for one or more tables, but a table can have only one full-text index.

Full-text catalogs are not stored in regular data files, but are still treated as files in the database are therefore included in the database file set you can back up. You can perform a complete (full) or a differential backup and restore of a full-text catalog. An individual full-text index created for a particular table can be assigned to a FileGroup and then backed up or restored as usual.

FILESTREAM Data

In SQL Server 2008, Microsoft introduced Filestream storage. Filestream data are often very large and unstructured objects that typically reside outside of a database, such as text documents, images, videos or music files. In SQL Server 2008, Filestream data is stored in separate FileGroups that contain only file system directories instead of the Filestream objects themselves.

Note: Large FILESTREAM collections may take significant amounts of time to estimate, causing property list generation to be slow.

Performance Improvement Tips

Consider the physical layout of your database to take advantage of the way files and FileGroups and partial backups/restores can optimize your database performance.

■ If you separate tables with high traffic estimates from tables with less expected traffic by placing them into different FileGroups, you can store the high-traffic table on high-performance disks and store the other files in another FileGroup on different disks. Queries against the a table and FileGroup backup jobs can then access your fastest devices.

■ Create Files and FileGroups on as many separate devices as possible so that queries against their tables can be processed by multiple read/write devices.

■ Put different tables used in the same queries in different FileGroups to enable parallel data searches.

■ Put the transaction log file on a disk that does not also contain data files.

■ If you are using Microsoft SQL Server 2005 or later, you can partition tables across multiple FileGroups to speed query access times (queries scan part of the data instead of all of it) and simplify tasks like rebuilding indexes. Consider horizontal or vertical partitions. See the Microsoft SQL Server documentation for more details.

■ Consider data stability to help you allocate files and FileGroups. For example, data that is static but needed for historical purposes can be assigned to a Read-only FileGroup. With SQL 2005 and later, use the Partial Database subset option to exclude Read-only FileGroups from your backup plan, improving backup time.

■ You can individually restore the files in a database in which multiple FileGroups are used. If a database is allocated on several devices and a disk fails, only the file on the failed disk must be restored.

■ Put the files expected to grow quickly in different FileGroups on separate drives.

■ As files become full, you can add files and disks to existing FileGroups, allowing SQL Server to pass data to the new file.

Partitioning

In SQL Server 2005, Microsoft introduced Partitioning, a method that makes large data sets easier to manage by limiting how much data is read during a single query. Partitioned tables can be divided among more than one FileGroup in a database. As you plan a partition scheme, determine the FileGroup or FileGroups you will put the partitions on. Assigning partitions to separate FileGroups ensures you can perform independent backup and restore operations. There are two types of table partitioning:

Horizontal Partitioning

Analyze your data for access trends. Partitioning a table horizontally means each FileGroup contains the same number of columns, but fewer rows. This is often done for tables containing data accumulated over time, so that a smaller window of time may be searched during a query.

Vertical Partitioning

Partitioning a table vertically means each FileGroup contains fewer columns, but the same number of rows. There are two types of vertical partitioning:

Normalization

This moves redundant columns from a table and stores them in smaller tables linked to it by a primary key relationship.

Row Splitting

This segments a table into smaller tables with fewer columns so that joining the nth row from each new table reassembles the row in the original table.

Database Mirroring

In SQL Server 2005, Microsoft introduced database mirroring, a method of increasing your database availability. With database mirroring, two copies of a database reside on computers in different locations, with one database in the active role and the other acting in the mirror role. Database mirroring is permitted on only databases using the full recovery model and is not permitted on the master, msdb or model databases.

Databases acting in the role of mirror are not eligible for backup. Therefore, Arcserve Backup does not show them unless they become "active". However, if you select the entire SQL Server for backup, mirrored databases in the active role are included by default, even if they served as the "mirror" when the job was created.

Recovery Models

The SQL Server recovery model is a decision that manages the risk of lost information in case of a disaster by controlling the level of transaction log involvement. You may change recovery models on a per-database basis to help you manage database maintenance tasks. Depending on the version of Microsoft SQL Server and the recovery model inherited by a database, certain Arcserve Backup backup options may not be available.

In a given SQL Server instance, your databases can have a mix of the following recovery models:

- **Simple --** Allows your database to be recovered only to the time of a backup. Transaction log backups are not permitted so any work done after the most recent backup must be redone. File and FileGroup backups are also not permitted, though in SQL 2000 and later, partial database backups are still permitted. The risk of lost information exists and is limited to all changes made since the last backup.

- **Full --** Allows your database to be recovered to the point of failure, or any point in time. Including transaction log backups is required so you can recover up to a specific point in time. File and FileGroup backups or database differential backups can be optionally included. This model has the lowest risk of data loss and the greatest flexibility during recovery.

- **Bulk-logged** -- Allows high-performance batch operations. Transaction log backups are required but you can recover only until the time of backup. You should perform transaction log backups to truncate the transaction log regularly. File and FileGroup backups or database differential backups can be optionally included.

Backup Method	Simple	Full	Bulk-Logged
Database Full	Required	Required	Required
Database Differential (not available for master db)	Optional	Optional	Optional
Transaction Log	Not Available	Required	Required
File and FileGroup (requires SQL Server 2000 or later)	Not Available	Optional	Optional
Partial Database (Requires SQL Server 2000 or later)	Optional	Optional	Optional
Backup Transaction Log After Database		Optional	Optional

For more information about File and FileGroup Backups, see Files and FileGroup Backups. (see page 119) For more information about Partial Database backups, see Partial Backups (see page 120).

System Databases

You can use Arcserve Backup to manage the protection and recovery of SQL Server system databases:

master database

The master database stores all system-level information for a SQL Server instance.

msdb database

The msdb database stores all information required by the automation agent service for SQL Server for scheduling alerts and jobs.

model database

The model database acts as a template for all databases created on a SQL Server instance. It stores metadata including database size, the recovery model, and other options that are inherited by each database you create.

Resource Database (SQL Server 2005 or later)

The Resource Database is read-only. It contains system objects required by SQL Server. It is not mounted as a live database, so the files are included in a file system backup rather than in a database backup.

tempdb database

The tempdb database holds memory-resident objects, or other temporary objects such as results needed for query operations.

distribution database

You may also have a Replication Distribution database, if your server is configured as a replication distributor. This database stores metadata and history information related to replication transactions and is sometimes referred to as the replication database.

Databases for extended SQL Server services

Additional databases may be created and used by certain features or secondary SQL Server components and services. These are handled the same as any production database in most cases.

Important! The three main system databases (master, msdb, and model) are exempt from the Global or Rotation Job backup method. Selecting this option for these databases always results in a Full backup.

Backup and Restore Considerations in Microsoft SQL Server Cluster Environments

Arcserve Backup supports clustered Microsoft SQL Server in the Microsoft Clustering Server (MSCS) environment.

Use Arcserve Backup and the Agent for Microsoft SQL Server to back up and restore clustered Microsoft SQL Servers just like any nonclustered instance of Microsoft SQL Server, with the following important differences:

- You must install the Agent for Microsoft SQL Server on the local drives of every node in the cluster.

  Note: For information about installing the Agent for Microsoft SQL Server, see Install the Agent in a Standard Microsoft SQL Server Environment.

- If the node on which the clustered Microsoft SQL Server instance is currently running fails over during a backup job, the backup job fails too, and you must restart the job if no makeup job is generated.

Using Microsoft SQL Server Writer on a Windows Server 2012 in AlwaysOn Availability Groups Environments

When using Microsoft SQL Server Writer for physical or virtual nodes on a Windows Server 2012 in AlwaysOn Availability Groups (SQL Server) environments, consider the following scenarios:

Physical-Active nodes:

- When the data for SQL Server instances is configured as AlwaysOn Availability Groups, the node follows the same behavior as that of non-cluster environments.

- The data for the SQL Server is visible. You can back up the data from the SQL Server writer from the Active Node.

Physical-Passive nodes:

- When the data for SQL Server instances is configured as AlwaysOn Availability Groups, the SQL instances become visible. However, you cannot back up the instances. Optionally, you can back up only the SQL Server instances that are not configured as AlwaysOn Availability Groups.

Virtual nodes:

- When the data for SQL Server instances is configured as AlwaysOn Availability Groups, the SQL Server instances become visible. However, you cannot back up the instances.

Best Practices for Backing Up AlwaysOn Availability Groups Using the Agent for Microsoft SQL Server

Review the following best practices for backing up AlwaysOn Availability Groups (AAG) using the Agent for Microsoft SQL Server on Windows Server 2012 operating systems:

- You do not need to define custom configurations to back up databases using the agent from Primary databases.

- Verify that all required, custom configurations are correct when backing up databases using the agent from Secondary, Replica databases by completing the following steps:

  1. Select and right-click the secondary AAG and click Properties on the pop-up menu to open the Availability Group Properties screen.
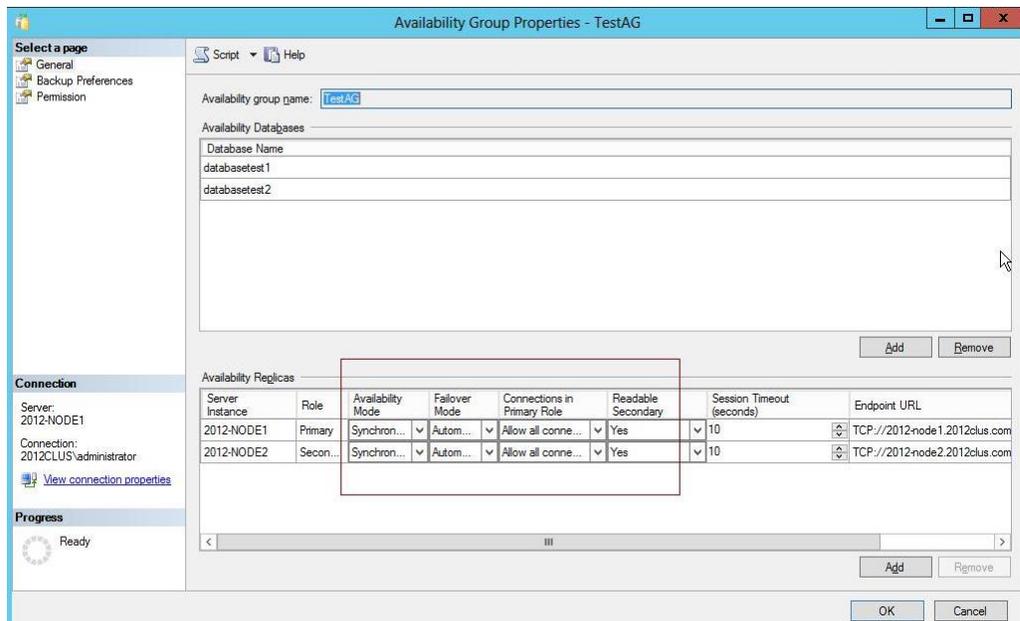
  2. Verify that the following values are defined for the AAG:

     – **Availability Mode:** Synchronous commit

     – **Readable Secondary:** Yes

  3. Click OK.

Backup Concepts

A *backup* is a copy or image of a database, Transaction Log, or collection of files or FileGroups that is stored on another device (typically, a media drive). A *differential backup* is a collection of changes or differences that have been made within a database and can be combined with an earlier backup to create a newer image. Use Arcserve Backup and the Agent for Microsoft SQL Server to generate backups using the SQL Server Backup statement.

To back up a database is to create a copy of its tables, data, and user-defined objects. In the event of media failure, you can recover your databases if you made regular backups of your databases and their Transaction Logs.

Important! Transaction Logs are not backed up or truncated during full or differential database backups. To back up and truncate Transaction Logs, perform a separate Transaction Log backup or use the Backup Transaction Log After Database option, and select the "Remove inactive entries from Transaction log" option to truncate the log files. For more information about Transaction Log backups, see Transaction Log Backups.

When a Microsoft SQL Server backup is started in Arcserve Backup, the Agent for Microsoft SQL Server initiates an online backup of a database or Transaction Log. This backup takes place while the database is active. The backup captures the state of the data at the moment the statement is executed. No partial transactions are captured. Any data changes made after the backup begins are not captured in the backed up copy of the database.

Backup Considerations

You should back up a database immediately after you create it and continue to back it up on a regular schedule to ensure smooth recovery from a database or media failure. Maintain regular backups of all databases, including:

- The master, msdb, and model databases

- All user databases and all SQL Server extended service databases

- The distribution database (if the server is configured as a replication distributor)

Important! Microsoft SQL Server 2000, 2005, 2008, and 2012 Mirror databases and Point-in-Time Snapshots cannot be backed up and do not appear in the database list. For more information about database mirroring and Point-in-Time Snapshots of databases, see the Microsoft SQL Server documentation.

During backup jobs, certain files are excluded. Arcserve Backup has two registry keys you can use to control the files to include or skip. The type of database agent you are using determines which registry key to use for this purpose. To include SQL database files in a backup job, set the BackupDBFiles registry key to 1. Do not use SkipDSAFiles with SQL database files. For more information, refer to the topic, Skip or Include Database Files in Backups (see page 85).

Important! Performing a backup can slow the system down. For better backup performance and to minimize impact on applications using the database, run backups when the database is not being heavily updated.

Required Full Backup

Restoring a database to a point in time or to the point of failure requires a specific restore sequence that begins with the last Full Database backup performed prior to that point. Any Differential, Partial or Transaction Log backups performed after the Full Backup are restored next, to roll the database forward. After you perform certain database managerial tasks, you must perform a new Full Database backup to re-establish the restore sequence. If you perform a Database Differential, Transaction Log, Partial Database or File and FileGroups backup after performing one of these tasks, a Full Database backup performed prior to performing the task may no longer work to successfully restore the database.

To ensure a successful database restore, always perform a Full Database backup immediately after you have performed any of the following actions:

- Created a new database

- Changed the recovery model of the database

- Changed the number of files or FileGroups in the database

- Changed the arrangement of files among the FileGroups

- Changed the arrangement of tables among the FileGroups

- Changed the partition scheme of a Partitioned Table or Index

- Canceled a full backup job while it is running

- Modified the database definition, including adding or removing a table, or changing the definition of a table

- Added or removed an index

- Restored the database from backups

Database Consistency Checks

When your database activity is low, you should run a database consistency check (DBCC), particularly with a large database. Although it takes some time, it is important to determine that your Microsoft SQL Server database is functioning well.

A DBCC tests the physical and logical consistency of a database. When you enable the Database Consistency Check option for a backup, the DBCC performs the following tests:

- **DBCC CHECKDB**: Checks the allocation and structural integrity of all objects in the specified database. By default, the CHECKDB performs a check for indexes that can increase the overall execution time.

- **DBCC CHECKCATALOG**: Checks for consistency in and between system tables in the specified database.

For more information, see Database Consistency Check (DBCC) Options.

Backup Strategy Recommendations

To establish a good backup strategy, follow these recommendations:

- If your database activity is low to medium, we recommend the following pattern:
  - Full backups: once per week
  - Differential backups: once per day
  - Transaction Log backups: every two to four hours

- If your database activity is high and your database is of small to medium size, we recommend the following pattern:
  - Full backups: twice per week
  - Differential backups: twice per day
  - Transaction Log backups: every 60 minutes

- If your database activity is high for a large size database using the Full or Bulk-Logged Recovery model, we recommend the following pattern:

    – Full backup: once per week

    – Differential backup: once per day

    – Transaction Log backup: every 20 minutes

- If your database activity is high for a large size database using the Simple Recovery model, we recommend the following pattern:

    – Full backup: once per week

    – Differential backup: twice per day

Full Backups

A Full Database backup creates a copy of the entire database at the time the backup command was executed, including all schema data, user data, Files and FileGroups and part of the Transaction Log, regardless whether that data has changed since the last backup. However, it is important to perform regular and separate Transaction Log backups in order to capture the entire Log file and truncate the log so it does not grow too large.

A Full backup can be ordered from the Backup Method selections of the Agent Options, from the Global Agent Options, or from the Rotation options in the Job Scheduler.

If you select a Full backup from the Agent Options dialog, you may refine the backup job by setting Database Subset options. For example:

- Combine a "Full" Backup Method with an "Entire Database" Database Subset selection to back up all schema, data, Files and FileGroups, regardless if data has changed.

- Combine a "Full" Backup Method with a "Files and FileGroups" Database Subset selection to back up all files from specific FileGroups in their entirety or to isolate specific data files. If you use FileGroups to separate Read-Only files from Read-Write files, this is one way to limit how often your Read-only FileGroups are backed up.

- Combine a "Full" Backup Method with a "Partial Database" Database Subset selection to back up all files in all Read/Write FileGroups, including the Primary FileGroup, regardless if data has changed.

Differential Backups

A differential backup records only the data that has changed since the last full backup of the same part of the data/same data files/same database subset. Usually, these backups are smaller and faster than full backups, but larger and longer than Transaction log backups. To restore a database, a differential backup requires only the last full backup as a prerequisite, and does not need any of the other differential or Transaction log backups performed between it and the last full backup. A differential backup is also faster to restore than a Transaction log backup because transactions do not need to be reprocessed.

Note: If a database is very active or if a long time has passed since the last full backup, a differential backup could take as long as a full backup.

For more information, see Differential Backup Timing (see page 117).

Differential Backup Timing

Perform differential backups as a supplement to full backups. Because they are usually faster and smaller, you can perform them more often than full database backups. They are also typically more efficient because they require less space on media and have a briefer impact on database performance than frequent full backups. Additionally, you can use them to minimize the number of Transaction logs you need to recover during a restore, because you need to restore only the Transaction log backups performed since the differential backup.

Differential backups are most beneficial under the following circumstances:

- A relatively small portion of the data in the database has changed since the last database backup. Differential database backups are most efficient if the same data is modified frequently.

- You are using the Simple Recovery model, which does not permit Transaction log backups, and you want to perform backups more frequently than is practical for full database backups.

- You are using the Full or Bulk-Logged Recovery model and want to minimize the time it takes to replay Transaction log backups when restoring a database.

Note: After you have modified the structure or configuration of the database (for example, by adding more data or log files or changing the recovery model), you must perform a full database backup before performing a differential or Transaction log backup. For more information, refer to Required Full Backups (see page 114).

Transaction Log Backups

Transaction logs record Microsoft SQL Server database activity. When using the Full or Bulk-Logged Recovery Models, they should be backed up frequently. To back them up, run self-contained Transaction log backups separately from database backups. Transaction log backups offer the following advantages over other types of backups:

- Generally faster than differential backups

- Typically faster and smaller than full database backups (unless they have not been truncated recently)

- Typically have less impact on database performance while running

- Normally able to be restored to a specified point in time, rather than only to the time the backup was made

After you have modified the structure or configuration of the database (for example, by adding more data or log files, or changing the recovery model), you must perform a full database backup before performing a differential or Transaction log backup. For more information, refer to Required Full Backup (see page 114).

It is sometimes possible to perform a Transaction Log backup of a damaged database. If the database is in the Suspect or Damaged state, and its transaction log files are intact, you will be able to perform a Transaction Log backup without Truncation. This will allow you to recover the database to the point in time immediately before the failure.

Important! Transaction logs are not backed up during full or differential database backups. You must back them up by running separate Transaction log backups or using the Backup Transaction Log After Database option. Transaction logs are only Truncated as part of a Transaction Log backup. If they are not backed up and truncated, they may grow until the disk is full. If this happens, you must perform a Transaction Log backup With Truncation, and then shrink the Transaction Log files to reclaim disk space. In SQL 2005 and later, multiple log backups with Truncation may be needed before log files can be significantly shrunk.

Note: Databases which use the Simple Recovery Model do not permit or require Transaction Log backups. Microsoft SQL Server manages the maintenance of the Transaction Logs for these databases automatically.

This section contains the following topics:

- Truncate Transaction Logs (see page 118)

Truncate Transaction Logs

You can truncate Transaction logs when you back them up. To truncate a Transaction log, select the Remove inactive entries from transaction log option when you configure the backup. If the Transaction log is not truncated, it may eventually grow large.

Files and FileGroup Backups

You can back up one or more FileGroups or individual files when the database size and performance requirements make it impractical to perform a full database backup.

If you back up an individual file instead of the full database, create procedures to ensure that all files in the database are backed up regularly, and perform separate Transaction log backups for the databases whose files or FileGroups you back up individually. After restoring a file backup, you must apply the successive Transaction log backups to roll the contents of the file forward to make it consistent with the rest of the database. For more information, see the Microsoft SQL Server documentation.

Microsoft SQL Server 2000 and later allow both Full and Differential File and FileGroup backups. If a File and FileGroup Differential backup is performed, restore of this session is dependent on the last Full backup of each file being restored. If you perform Full and Differential backups with different file sets but which contain some files in common, Arcserve will not be able to construct the correct dependency sequence to restore these backups, and you will need to do so manually.

Important! In Microsoft SQL Server 7.0 and 2000, a Database Differential backup is not dependent on any Files-and-FileGroups Full backups that might be between it and the previous Database Full backup. However, in SQL Server 2000 and later, a Database Differential or Partial Differential backup is dependent on any intervening Full backups, including Files-and-FileGroups Full and Partial Full backups. The Automatic Selection feature cannot detect this condition. If you mix Files-and-FileGroups backups with Database Differential and Partial Differential backups in SQL Server 2005, you may need to assemble the restore sequences manually.

Note: You cannot perform Database or Partial restores from a Files-and-FileGroups backup. Only a Files-and-FileGroups restore can be performed from a Files-and-FileGroups backup.

How the Create Index Statement Impacts File and FileGroup Backups

The Backup statement requires that you back up entire FileGroups affected by a Create Index statement. This requirement exists in the following situations:

- If you create an index on a FileGroup, you must back up that entire FileGroup in a single backup operation. Microsoft SQL Server does not allow backups of individual files that are part of the affected FileGroup.

- If you create an index on a FileGroup separate from the FileGroup in which the table resides, then you must backup both FileGroups (the FileGroup containing the table and the FileGroup containing the newly created index) together.

■   If you create more than one index on a FileGroup separate from the FileGroup in which the table resides, you must back up all the FileGroups immediately to accommodate these different FileGroups.

The Backup statement detects all of these FileGroup situations and communicates the minimum number of FileGroups that you must back up. Microsoft SQL Server reports this information when the backup job is run in one or more error messages, which will be written to the Activity Log by the Agent for Microsoft SQL Server.

Partial Backups

Microsoft SQL Server 2005 introduces a special type of Files-and-FileGroups backup called a Partial Backup. These backups automatically select the Primary FileGroup, along with all other FileGroups that are not Read-Only. If the database itself is Read-Only, then only the Primary FileGroup will be included.

Unlike Files-and-FileGroups backups, Partial Backups can be performed on databases that use the Simple Recovery Model, because all of the data that can be changed is included. If you have a large database that contains a large amount of static data in Read-Only FileGroups, you can use a Partial Backup to reduce the size and time required to perform regular backups. As long as you retain the latest Full Database backup and the database structure does not change, you can use Partial Full and Partial Differential backups without needing to perform another Full Database backup.

Also, you can perform a Partial Restore from a Partial Backup, provided all of the desired FileGroups are contained within the Partial Backup session. You can also perform a Database Restore from a Partial Backup, with the last Database Full backup as a pre-requisite.

Note: When using the Simple Recovery Model, you cannot perform a Database Differential backup after a Partial Full backup, until you first perform another Database Full backup. For all recovery models, if you perform a Partial Full backup after a Database Differential backup, the Partial Full backup will be dependent on the Database Differential backup as a pre-requisite.

Important! In SQL Server 7.0 and 2000, a Database Differential backup is dependent on any Files and FileGroup backups that might be between it and the previous Database Full backup. However, in SQL Server 2005 and later, a Database Differential or Partial Differential backup is dependent on any intervening Full backups, including Files and FileGroups Full and Partial Full backups. The Automatic Selection feature cannot detect this condition. If you mix Files and FileGroup backups with Database Differential and Partial Differential backups in SQL Server 2005, you may need to assemble the restore sequences manually.

Restore Overview

To *restore* is to load a database from a backup of that database and (if applicable) one or more backups of its Transaction log. If a database is lost or damaged, you can restore the database by reloading the most recent database backup and the successive log backups. A restore overwrites any information in the database with the backed up information. Use Arcserve Backup and the Agent for Microsoft SQL Server to perform restore operations using the Microsoft SQL Server Restore statement.

When you restore a database that is live and online, Microsoft SQL Server rolls back any uncommitted transactions that were active at the moment the restore job began. When the restore operation is complete, the database is in the same state it was in when the Backup statement for the backup used in the restore job was initiated, excluding any transactions that were active at that point.

As the data from the backup is being restored, Microsoft SQL Server reinitializes any remaining unused pages. For example, if a 100 MB database contains only 5 MB of data, Microsoft SQL Server rewrites all 100 MB of space. Consequently, it takes at least as long to restore a database as it does to create a database.

Microsoft SQL Server locks a database while restoring it, so that the database cannot be modified during the restore operation. However, users can access and modify other Microsoft SQL Server databases during this time.

Note: If a restore is attempted when you are accessing a database, Microsoft SQL Server will not allow the restore operation.

Important! By default, Microsoft SQL Server 2005 will not allow a restore to be attempted if a database is online and is using the Full or Bulk-Logged Recovery Models. To restore these databases, you must either take the database offline by performing a Transaction Log backup with the Log Tail option so that no transactions are lost, or restore with the Overwrite Existing Database and Files ("WITH REPLACE") option. The only exception to this is an Online Torn Page Repair restore.

If a failure occurs while a database is being restored, Microsoft SQL Server notifies the system administrator but does not recover the partially restored database. You must restart the database restore to complete the restore job.

Note: If you cancel a restore job, the database may be left in a loading state and will be unusable until the restore sequence is completed. If the session that was being restored when the job was cancelled is not the first session in the restore sequence, you may have to start the restore sequence over from the beginning.

The destination database must have at least as much storage space available for it as the backed up database. The actual amount of data in the backed up database is not relevant for this requirement. To get information about allocated storage space, use the Microsoft SQL Enterprise Manager or Management Studio, or the DBCC CHECKALLOC statement. If the database is offline, then check the file sizes listed in the file tree of the Agent Restore Options dialog for that session, or in the Arcserve Database Manager view.

If you have a media failure, restart Microsoft SQL Server. If, after a media failure, Microsoft SQL Server cannot access a database, it marks the database as Suspect, locks it, and displays a warning message. You may have to drop (detach from Microsoft SQL Server) a damaged database, a process that you can perform using the Microsoft SQL Server Enterprise Manager or Management Studio.

Restore Types and Methods

You can restore from the following types of backup sessions:

- Database Full and Differential backups

- Partial Full and Differential backups

- Transaction log backups

- File and FileGroup Full and Differential backups

For any type of restore operation, you can use the following restore methods:

- **Restore by Tree:** This option displays a tree of the computers and objects backed up by Arcserve Backup. To run a restore operation, expand the appropriate computers and instances to select the databases you want to restore. The databases are from the most recent backup sessions. This option is selected by default.

- **Restore by Session:** This option displays a list of media you have used when backing up with Arcserve Backup. To run a restore operation, expand the media containing the backup you want, and select the session containing the specific database or log session you want to restore.

Note: Arcserve Backup supports only Restore by Tree and Restore by Session for Microsoft SQL Server databases.

Checklists for Dependencies, by Type

Using the Restore by Tree or Restore by Session method, you can perform various types of restores. Each type has specific dependencies and requirements for a successful restore. The following tables organize the requirements according to restore type.

Database Restore Dependency Checklist

A Database Restore restores and recovers the database in its entirety. You can restore a database from a Database Full, Database Differential, Partial Full or Partial Differential Backup.

Selected Session	Prerequisites	Post-requisites
Database Full Backup	None	None
Database Differential Backup For more information, see Note 1 (see page 128).	The last prior Database Full backup	None
Partial Full Backup	The last prior Database Full backup	None
Partial Differential Backup For more information, see Note 1 (see page 128).	The more recent of the following: ■ The last prior Database Full backup ■ The last prior Partial Full backup	None

Transaction Log Restore Dependency Checklist

A Transaction Log Restore restores and processes the Transaction Log contents, a process also known as "applying" or "replaying" the logs. Microsoft SQL Server re-executes the changes contained in the log and rolls back any transactions that were uncommitted when you backed up the Transaction Log. Applying Transaction Logs lets you recover as much of a database as possible, up to the time when the last Transaction Log backup was performed. The only uncommitted transaction that Microsoft SQL Server does not roll back is the Microsoft SQL Log Backup transaction, which is completed as part of the restore process instead. Transaction Log Restores can only be performed from Transaction Log Backups.

Selected Session	Prerequisites	Post-requisites
Transaction Log Backup	Any of the following:	None, if selected as a Transaction Log restore
	■ The last prior Transaction Log Backup	The next Transaction Log Backup, if included as part of a Files and FileGroups or Torn Page Repair restore
	■ The last prior Transaction Log with Truncation Backup	
	■ The last prior Database Backup	
	■ The last prior Partial Backup	
	■ Or, the selected session for Partial, Files and FileGroups, or Torn Page Restore	

Note: If the last backup before a Transaction Log backup with Truncation was a Transaction Log backup, that session specifically is the only required prerequisite.

Files and FileGroups Restore Dependency Checklist-MSSQLSvrWSPW

A Files and FileGroups Restore restores and recovers selected files and FileGroups. You can restore Files and FileGroups from a Files and FileGroups, Partial, or Database backup.

A Files and FileGroups restore, like a Differential or Log restore, can only be applied to either the original database from which it was backed up, or to a copy that has never left the loading or standby state. A Mirror database qualifies as either a standby copy or the original database for this purpose.

Sequence is important when performing a Files and FileGroups restore.

Selected Session	Prerequisites	Post-requisites
Database Full Backup	None For more information, see Note 2 (see page 128).	Use one of the following: ■ All successive Transaction log backups ■ All successive Transaction Log backups with Truncation and Transaction Log Tail backups
Partial Full Backup	None For more information, see Note 2 (see page 128).	Use one of the following: ■ All successive Transaction log backups ■ All successive Transaction Log backups with Truncation and Transaction Log Tail backups
Files and FileGroups Full Backup	None For more information, see Note 2 (see page 128).	Use one of the following: ■ All successive Transaction Log backups ■ All successive Transaction Log backups with Truncation and Transaction Log Tail backups
Database Differential Backup For more information, see Note 1 (see page 128).	The last prior Database Full backup	Use one of the following: ■ All successive Transaction Log backups ■ All successive Transaction Log backups with Truncation and Transaction Log Tail backups

Selected Session	Prerequisites	Post-requisites
Partial Differential Backup For more information, see Note 1 (see page 128).	Use the more recent of the following: ■ The last prior Database Full backup ■ The last prior Partial Full backup	Use one of the following: ■ All successive Transaction Log backups ■ All successive Transaction Log backups with Truncation and Transaction Log Tail backups
Files and FileGroups Differential Backup For more information, see Note 1 (see page 128).	The last prior Full Backup that contains the files included in the selected session	Use one of the following: ■ All successive Transaction Log backups All successive Transaction Log backups with Truncation and Transaction Log Tail backups

Partial Restore Dependency Checklist

A Partial Restore restores and recovers the Primary FileGroup and any other FileGroups you specify in the Restore Options as a new database. The result is a subset of the database. FileGroups that are not restored are marked as offline and are not accessible.

Due to some FileGroups being offline, a database that was created from a Partial Restore may not be eligible for a Database backup. To resolve this condition, either restore the remaining FileGroups to the partially-restored database, or remove them from the database structure using the Microsoft SQL Server Enterprise Manager or Management Studio.

Selected Session	Prerequisites	Post-requisites
Database Full Backup	None	None
Partial Full Backup	None	None
Database Differential Backup	The last prior Database Full Backup	None
Partial Differential Backup For more information, see Note 1 (see page 128).	The more recent of the following: ■ The last prior Database Full Backup ■ The last prior Partial Full Backup	None

Torn Page Repair Dependency Checklist

A Torn Page Restore restores only the pages of data on disk marked as damaged. SQL Server 2005 and later have the ability to isolate damaged parts of a database while leaving the rest intact. When this happens with a Full Recovery Model database, the database may be eligible for a Torn Page repair, which is much faster than a Database restore. The dependencies for this restore type are similar to those for a Files and FileGroups restore.

There are both Online and Offline versions of this operation.

- Online -- The database remains online and undamaged tables are still accessible. This requires the Enterprise Edition of SQL Server.

- Offline -- The database should be taken offline using a Log Tail backup before the restore is performed.

Selected Session	Prerequisites	Post-requisites
Database Full Backup	None For more information, see Note 2 (see page 128).	Use one of the following: - All successive Transaction log backups - All successive Transaction Log backups with Truncation and Transaction Log Tail backups
Partial Full Backup	None For more information, see Note 2 (see page 128).	Use one of the following: - All successive Transaction log backups - All successive Transaction Log backups with Truncation and Transaction Log Tail backups
Files and FileGroups Full Backup	None For more information, see Note 2 (see page 128).	Use one of the following: - All successive Transaction Log backups - All successive Transaction Log backups with Truncation and Transaction Log Tail backups
Database Differential Backup For more information, see Note 1 (see page 128).	The last prior Database Full backup	Use one of the following: - All successive Transaction Log backups - All successive Transaction Log backups with Truncation and Transaction Log Tail backups
Partial Differential Backup For more information, see Note 1 (see page 128).	Use the more recent of the following: - The last prior Database Full backup - The last prior Partial Full backup	Use one of the following: - All successive Transaction Log backups - All successive Transaction Log backups with Truncation and Transaction Log Tail backups

Selected Session	Prerequisites	Post-requisites
Files and FileGroups Differential Backup For more information, see Note 1 (see page 128).	The last prior Full Backup that contains the files included in the selected session	Use one of the following: ■ All successive Transaction Log backups ■ All successive Transaction Log backups with Truncation and Transaction Log Tail backups

Note 1

In SQL Server 2005 and later, a Differential backup of any particular data file is dependent on the last prior Full backup of that data file. If a data file is contained in a Partial Full or Files-and-FileGroups Full backup, a Database Differential backup made after that Partial Full or Files-and-FileGroups backup and before the next Database Full backup will be dependent on that Partial Full or Files-and-FileGroups backup. The same applies to a data file included in a Files-and-FileGroups Full backup and a Partial Differential backup, or if the file selections vary between File-and-FileGroup Full and File-and-FileGroup Differential backups. Arcserve Backup does not detect these conditions in this release.

Alternatively, restoring Transaction Log backups that are between the Database Full or Partial Full backup and the Database or Partial Differential backup will serve to place the database in a correct state to restore from the Differential backup. However, this will take longer to execute than restoring the intervening Files-and-FileGroups Full or Partial Full backups.

Note 2

To perform a Files-and-FileGroups restore or an Offline Torn Page Repair, the database must be in a Loading or standby state. This is usually accomplished by performing a Transaction Log Tail backup but may also be accomplished by performing a Database restore of the selected session or older using the option, Leave database non-operational but able to restore additional transaction logs. However, if you restore an older session, you will need to restore all Transaction Log sessions between the older session and the selected session, or the Transaction Log restores after the Files-and-FileGroups restore will not be able to be applied to the other database files, and may fail.

Microsoft SQL Server Disaster Recovery Elements

When you backup a Microsoft SQL Server instance that includes Full Database backups of the master, model and msdb system databases, the Agent for Microsoft SQL Server will generate an additional backup session called Microsoft SQL Server Disaster Recovery Elements. This session contains an image of the master and model databases that can be restored as files, with the Microsoft SQL Server instance offline. Restoring this session will take the place of performing a Rebuild master Database operation. This allows you to bring Microsoft SQL Server online, allowing you to restore from backups that have been taken online.

The Disaster Recovery Elements session will appear in the Restore by Tree view with the name "Microsoft SQL Server Disaster Recovery Elements", followed by the Instance Name for a named instance of Microsoft SQL Server. In the Restore by Session view, it will have a volume name of "sqldr@" followed by the instance name. For a default instance of Microsoft SQL Server, the instance name will appear as "MSSQLSERVER", regardless of the SQL Server version.

Restoring a Microsoft SQL Server Disaster Recovery Elements session to its original location will restore the files to the location where the master and model database files existed. Restoring them to an Alternate Location and selecting a disk or directory will place the files in the directory that you select. Restoring them to an Alternate Location and selecting only the computer will restore them to the original file path on the selected computer.

Note: If you attempt to restore SQL Server Disaster Recovery Elements to their original location while the database is online, the restore will fail because Microsoft SQL Server is using the existing files.

After restoring Disaster Recovery Elements, you should immediately restore the master database, the msdb database if it is offline, and the model database, from the regular online backups.

Important! Disaster Recovery Elements are specific to the instance that they were backed up from. If you use them for a different instance, that instance may not start, or may behave incorrectly after starting.

Note: The Arcserve Backup Disaster Recovery Option will include the Microsoft SQL Server Disaster Recovery Elements sessions when performing a Disaster Recovery restore. If you use the Disaster Recovery Option to restore the computer that houses the Arcserve Database, you should restore the three system databases first after the Recover Database operation is finished. For more information, see the *Administration Guide*.

Restore Master Databases

Before you restore the master database, if the database files are missing, you may need to either restore Microsoft SQL Server Disaster Recovery Elements for that instance, or rebuild it using the Microsoft SQL Server Rebuild Master utility. Restoring the master database requires that the restore operation have exclusive access to the SQL Server instance. To ensure this, the Microsoft SQL Server instance must be running in single-user mode. For complete instructions on rebuilding your Microsoft SQL Server master database, see the Microsoft SQL Server documentation.

Before restoring the master database, you must stop all applications which may be using that instance of Microsoft SQL Server. This includes some secondary services which are part of Microsoft SQL Server, such as the SQL Server Automation Agent ("SQL Server Agent" service). The only exception to this is Arcserve Backup itself; the Agent for Arcserve Database will ensure Arcserve Backup is not using any ARCserve Database that the instance might contain.

When you restore the master database, the Agent for Microsoft SQL Server automatically detects this. The agent will restart Microsoft SQL Server in Single-User Mode before executing the restore operation, and will return Microsoft SQL Server to Multi-User Mode at the conclusion of the master database restore.

Important! You must stop *all* applications and services other than Arcserve Backup that may be using databases in the Microsoft SQL Server instance for which you are restoring the master database. Otherwise, one of these other applications may re-connect to the SQL Server instance after it is restarted, and prevent the agent from performing the restore.

Glossary

Agent Options

Agent Options let you specify backup job options on a per-database basis or define a set of default options for all databases.

Database Consistency Check Options

These options test the physical and logical consistency of a database.

Database Level Agent Options

Options you apply to only the selected database to extend or override Global Option/Agent Options.

Database Subset Options

Database subset options let you define the database components you want to protect, i.e., the entire database, or files and FileGroups.

Dynamic Job Packaging

A method of packaging backup jobs that automatically selects for backup all the components of the selected object, i.e., child volumes and files.

Explicit Job Packaging

A method of backup job packaging that lets you pick the objects for backup individually, without regard for parent objects.

Global Options Agent Options

Global Options/Agent Options let you set default job options for all SQL Server Agent objects.

Universal Agent Service

Service shared with Arcserve Backup Agents, including the Agent for Microsoft SQL Server, that provides a single point of access for backup and restore operations. The service dynamically recognizes new agents as they are installed.

Index