



EMC[®] Avamar[®] 7.0 for SQL Server

User Guide

P/N 300-015-233
REV 03

Copyright © 2001 - 2013 EMC Corporation. All rights reserved. Published in the USA.

Published November, 2013

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to the technical documentation and advisories section on the EMC online support website.

CONTENTS

Preface

Chapter 1

Introduction

Architecture	16
Avamar components	16
How Avamar connects to SQL Server	17
How Avamar works in a high availability environment	17
Data Domain system support	26
Server virtualization	26
Backup	27
Backup types	27
Choosing a backup type based on the recovery model	28
Backups in AlwaysOn availability groups.....	29
Mixed backups.....	30
Multi-streaming.....	30
Restore	31
Restore to the original location.....	31
Restore to a new database in the original instance.....	32
Restore to a different instance on the original server	32
Restore to an instance on a different server.....	32
Restore to a file.....	33
Restore to an AlwaysOn availability group	33
Restore of a database with an intact log file	35
Restore of system databases.....	35
Tail-log backup	36
Point-in-time restore	37
SQL Server recovery operations.....	37
Disaster recovery	38

Chapter 2

Installation

Preparing to install the SQL Server plug-in.....	40
Checking the system requirements.....	40
Checking the SQL Server requirements.....	42
Checking the User Account Control setting on Microsoft Windows	46
Preparing a stand-alone server.....	46
Preparing a cluster	46
Downloading the software.....	47
Verifying the environment	48
Installing the Avamar client software.....	49
Installation road maps	49
Installing the Avamar Client for Windows.....	50
Installing the Avamar Plug-in for SQL Server	51
Registering the client	52
Configuring the cluster client in a failover cluster	52
Configuring the cluster client for an AlwaysOn availability group	54
Upgrading the Avamar client software	56
Upgrading on a stand-alone server.....	56
Upgrading in a cluster	57

Uninstalling the Avamar client software.....	58
Uninstall road map.....	58
Uninstalling the cluster client in a failover cluster.....	58
Uninstalling the cluster client for an AlwaysOn availability group	59
Uninstalling the Avamar Plug-in for SQL Server.....	60
Uninstalling the Avamar Client for Windows	60

Chapter 3

Backup

Performing on-demand backups.....	62
Selecting SQL Server data for an on-demand backup.....	65
Performing scheduled backups	67
Creating a dataset	67
Creating a group.....	74
Enabling scheduled backups	78
Setting backup options	79
Backup label option	79
Backup type options	80
AlwaysOn availability group option	82
Enhanced data deduplication option.....	83
Truncate database log option	83
Simple recovery model databases option	84
Enable debugging messages.....	84
Store backup on Data Domain system option	84
Multi-streaming options	85
Authentication options.....	92
Monitoring backups	93
Troubleshooting backups	94
Error appears for a forced incremental backup after a restore	94
Canceling backups	94

Chapter 4

Restore

Restore requirements.....	96
Software requirements for restore	96
Tail-log backup and point-in-time restore requirements.....	96
Requirements to restore secondary database files	97
Requirements to restore the Report Server database	98
SQL Server write permission requirements	98
Finding a backup.....	98
When to find a backup by date	98
How to find a backup by date	98
When to find a backup for an instance or database	101
How to find a backup for an instance or database	102
Restoring to the original location.....	106
Options when restoring to the original location	106
How to restore to the original location	106
Restoring to a new database in the original instance	109
Restoring to a different instance.....	113
Restoring to a file	119
Restoring to a file with the SQL Server plug-in	119
Restoring to a file with the Windows File System plug-in.....	123
Restoring a database with SQL Server tools.....	126
Restoring system databases.....	132
Restoring system databases automatically.....	132

	Restoring system databases manually	135
	Restoring to an AlwaysOn availability group	138
	Restoring to the original availability group.....	138
	Restoring to a new database in the original availability group	139
	Restoring to a different availability group	140
	Restoring to a SQL Server instance without availability groups	140
	Restoring to operating system files.....	141
	Restoring a database with an intact log file	141
	Setting restore options.....	142
	General restore options.....	143
	Recovery operation options.....	145
	System database restore options	146
	Redirected restore options	146
	Authentication options.....	147
	Point-in-time recovery options.....	148
	Monitoring restores.....	149
	Canceling restores	150
Chapter 5	Disaster Recovery	
	Preparing for disaster recovery	152
	Performing disaster recovery	152
Appendix A	Plug-in Options	
	How to set plug-in options	154
	Backup options.....	155
	Restore options.....	158
Appendix B	Command Line Interface	
	Understanding the SQL Server plug-in CLI	164
	CLI architecture	164
	Launching the CLI.....	164
	Available operations	164
	Options.....	165
	Using the CLI in a failover cluster.....	166
	Using the CLI in an AlwaysOn availability group.....	166
	Password encoding.....	167
	Help	167
	Monitoring CLI activities.....	167
	Command reference	168
	Synopsis	168
	Operations.....	168
	Options.....	168
	CLI examples	180
	Example browse commands.....	180
	Example backup commands.....	180
	Example restore commands	184
Glossary		

TABLES

	Title	Page
1	Revision history	10
2	Recovery operation options	37
3	Minimum and recommended hardware requirements.....	40
4	Avamar server and SQL Server plug-in compatibility	41
5	Availability group replica for backup option values.....	82
6	Enhanced data deduplication option values.....	83
7	Truncate database log option values	83
8	For simple recovery model databases option values.....	84
9	Recommended minimum stream size by backup set and environment	89
10	Examples of minimum stream size in mixed environments	91
11	Method and steps to browse to a backup by instance or database	103
12	Recovery operation options	145
13	Backup plug-in options	155
14	Restore plug-in options	158
15	Operations for the avsql command	168
16	Common avsql options.....	168
17	Account options for the avsql command	169
18	Logging options for the avsql command	170
19	Backup options for the avsql command	171
20	Restore options for the avsql command.....	174

PREFACE

As part of an effort to improve its product lines, EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your EMC technical support professional if a product does not function properly or does not function as described in this document.

Note: This document was accurate at publication time. Go to EMC Online Support (<https://support.emc.com>) to ensure that you are using the latest version of this document.

Purpose

This guide describes how to install Avamar in a Microsoft SQL Server database environment, and how to back up and restore SQL Server databases.

Audience

The information in this guide is primarily intended for:

- ◆ System administrators who are responsible for installing software and maintaining servers and clients on a network
- ◆ Microsoft SQL Server administrators who are responsible for backing up and maintaining Microsoft SQL Servers

Persons using this guide should have current practical experience with the following topics:

- ◆ Operating system shell commands on the SQL Server platform (root permission required)
- ◆ The specific version of Microsoft SQL Server currently deployed at the site

Revision history

The following table presents the revision history of this document.

Table 1 Revision history

Revision	Date	Description
03	November 25, 2013	Removed the compatibility table with information on supported SQL Server and operating system versions. Updated compatibility information is available in the <i>EMC Avamar Compatibility and Interoperability Matrix</i> .
02	August 19, 2013	<ul style="list-style-type: none"> Added a notice to “Upgrading the Avamar client software” on page 56 that a full backup is required after upgrade when you store backups on a Data Domain system. Removed text about support for restoring from a different SQL Server version. The target SQL Server installation for a restore must have the same SQL Server version and service pack as the SQL Server installation on which the backup occurred. Otherwise, restore fails.
01	July 10, 2013	Initial release of Avamar 7.0.

Related documentation

The following EMC publications provide additional information:

- ◆ *EMC Avamar Compatibility and Interoperability Matrix*
- ◆ *EMC Avamar Release Notes*
- ◆ *EMC Avamar Administration Guide*
- ◆ *EMC Avamar for Windows Server User Guide*
- ◆ *EMC Avamar and EMC Data Domain System Integration Guide*
- ◆ *EMC Avamar Operational Best Practices*
- ◆ *EMC Avamar for Hyper-V VSS User Guide*
- ◆ *EMC Avamar for VMware User Guide*

Conventions used in this document

EMC uses the following conventions for special notices:

NOTICE

NOTICE is used to address practices not related to personal injury.

Note: A note presents information that is important, but not hazard-related.

IMPORTANT

An important notice contains information essential to software or hardware operation.

Typographical conventions

EMC uses the following type style conventions in this document:

Bold	Use for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks)
<i>Italic</i>	Use for full titles of publications referenced in text
Monospace	Use for: <ul style="list-style-type: none"> • System output, such as an error message or script • System code • Pathnames, filenames, prompts, and syntax • Commands and options
<i>Monospace italic</i>	Use for variables.
Monospace bold	Use for user input.
[]	Square brackets enclose optional values
	Vertical bar indicates alternate selections — the bar means “or”
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

Where to get help

The Avamar support page provides access to licensing information, product documentation, advisories, and downloads, as well as how-to and troubleshooting information. This information may enable you to resolve a product issue before you contact EMC Customer Support.

To access the Avamar support page:

1. Go to <https://support.EMC.com/products>.
2. Type a product name in the **Find a Product** box.
3. Select the product from the list that appears.
4. Click the arrow next to the **Find a Product** box.
5. (Optional) Add the product to the **My Products** list by clicking **Add to my products** in the top right corner of the **Support by Product** page.

Documentation

The Avamar product documentation provides a comprehensive set of feature overview, operational task, and technical reference information. Review the following documents in addition to product administration and user guides:

- ◆ Release notes provide an overview of new features and known limitations for a release.
- ◆ Technical notes provide technical details about specific product features, including step-by-step tasks, where necessary.
- ◆ White papers provide an in-depth technical perspective of a product or products as applied to critical business issues or requirements.

Knowledgebase

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

To search the EMC Knowledgebase:

1. Click the **Search** link at the top of the page.
2. Type either the solution number or keywords in the search box.
3. (Optional) Limit the search to specific products by typing a product name in the **Scope by product** box and then selecting the product from the list that appears.
4. Select **Knowledgebase** from the **Scope by resource** list.
5. (Optional) Specify advanced options by clicking **Advanced options** and specifying values in the available fields.
6. Click the search button.

Online communities

Visit EMC Community Network at <http://community.EMC.com> for peer contacts, conversations, and content on product support and solutions. Interactively engage online with customers, partners and certified professionals for all EMC products.

Live chat

To engage EMC Customer Support by using live interactive chat, click Join Live Chat on the Service Center panel of the Avamar support page.

Service Requests

For in-depth help from EMC Customer Support, submit a service request by clicking Create Service Requests on the Service Center panel of the Avamar support page.

Note: To open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

To review an open service request, click the Service Center link on the Service Center panel, and then click View and manage service requests.

Facilitating support

EMC recommends that you enable ConnectEMC and Email Home on all Avamar systems:

- ◆ ConnectEMC automatically generates service requests for high priority events.
- ◆ Email Home emails configuration, capacity, and general system information to EMC Customer Support.

Your comments

Your suggestions help us to continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to:

BSGDocumentation@emc.com

Please include the following information:

- ◆ Product name and version
- ◆ Document name, part number, and revision (for example, 01)
- ◆ Page numbers
- ◆ Other details that will help us address the documentation issue

CHAPTER 1

Introduction

The following topics provide an introduction to using EMC® Avamar® to back up and restore data in a Microsoft SQL Server environment:

- ◆ Architecture 16
- ◆ Backup 27
- ◆ Restore 31
- ◆ Disaster recovery..... 38

Architecture

The following topics explain how Avamar works in a SQL Server environment:

- ◆ “Avamar components” on page 16
- ◆ “How Avamar connects to SQL Server” on page 17
- ◆ “How Avamar works in a high availability environment” on page 17
- ◆ “Data Domain system support” on page 26
- ◆ “Server virtualization” on page 26

Avamar components

The following figure illustrates a basic system architecture, including required Avamar software components, when you use Avamar to back up a SQL Server environment.

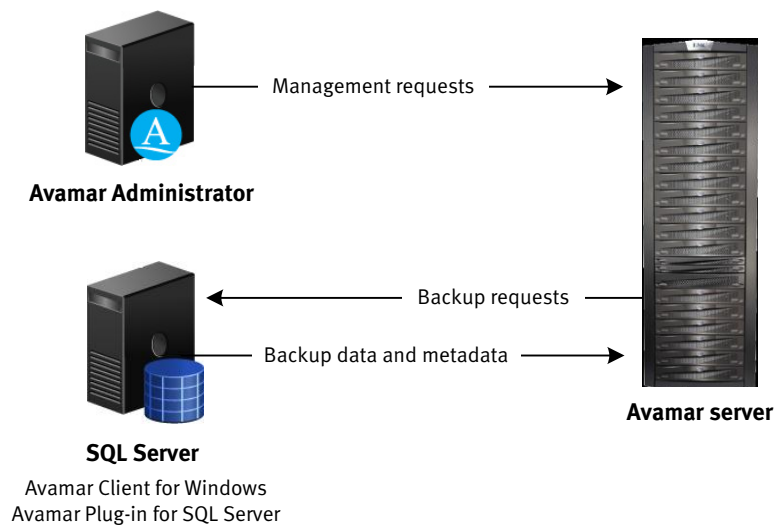


Figure 1 Avamar architecture with SQL Server

Install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on the computer that is running SQL Server:

- ◆ **Avamar Client for Windows** — When you install the Avamar Client for Windows, the installation includes the Avamar Plug-in for Windows File System and the Avamar agent, which is required for the Avamar Plug-in for SQL Server. You can use the Windows File System plug-in to back up operating system and SQL Server binary files, which are required for disaster recovery.
- ◆ **Avamar Plug-in for SQL Server** — The SQL Server plug-in enables you to back up and restore SQL Server instances and databases.

Avamar Administrator is a graphical management console software application for remote administration of an Avamar system from a supported Windows or Linux client computer. You can configure, perform, monitor, and manage backups and restores using Avamar Administrator. The *EMC Avamar Administration Guide* provides complete instructions for installing and using Avamar Administrator.

How Avamar connects to SQL Server

The Avamar Plug-in for SQL Server uses the SQL Server virtual device interface (VDI) to back up and restore databases and transaction logs.

Backup process overview

1. The SQL Server plug-in creates one or more VDI devices, and then executes the necessary SQL commands to back up the database to the VDI devices.

The number of VDI devices that the SQL Server plug-in creates depends on the number of data streams that you configure for a backup. Each data stream corresponds to a VDI device. [“Multi-streaming” on page 30](#) provides details on streams.
2. The Avamar `avtar` program is spawned in a standard input/output (stdio) mode.
3. The SQL Server plug-in reads data read from the VDI device and writes data to `avtar` standard input.
4. The `avtar` program reads and sends the data to either the Avamar server or a configured Data Domain system.

Restore process overview

During a restore, the SQL Server plug-in reads data from standard output and writes the data to the VDI device, which communicates with SQL Server to restore the databases.

Authentication

When Avamar connects to SQL Server for backup or restore, you can use either Windows authentication or SQL Server authentication:

- ◆ With Windows authentication, Avamar connects to SQL Server by using a Windows user account that has privileges in SQL Server. The account must be a domain account with the proper administrative privileges.
- ◆ With SQL Server authentication, Avamar connects to SQL Server by using a SQL Server login account. You must select the Mixed authentication mode for the Database Engine when you configure SQL Server.

How Avamar works in a high availability environment

Avamar supports the following high availability strategies for SQL Server data:

- ◆ High availability at the instance level through SQL Server Failover Cluster Instances (FCIs) in a Windows Server Failover Clustering (WSFC) cluster
- ◆ High availability of groups of databases through SQL Server 2012 AlwaysOn availability groups in a WSFC cluster
- ◆ High availability of an individual database through database mirroring

Failover Cluster Instances

You can install SQL Server in a Windows Server Failover Clustering (WSFC) cluster with two or more nodes. With Failover Cluster Instances (FCIs), SQL Server databases and log files are stored on a drive that is shared between the nodes. If there is a failure on one of the nodes, then the applications *fail over* to and run on another node. When the failed node comes back online, you can *fail back* the applications from the other node.

SQL Server can run as either active/passive or active/active in a cluster.

Active/passive cluster environment

In an active/passive configuration, SQL Server runs on the active node, which is the primary node. The passive node is a standby node. SQL Server does not run on the passive node. If a failure occurs on the primary active node, then SQL Server fails over to the passive node. The passive node becomes the active node. When the primary active node comes back online, you can fail back SQL Server from the standby node to the primary active node. The passive node returns to its standby role.

Avamar can perform both on-demand and scheduled backups of SQL Server data while SQL Server is running on either the primary active node or on a standby node.

The following figure illustrates an active/passive cluster environment with SQL Server and Avamar.

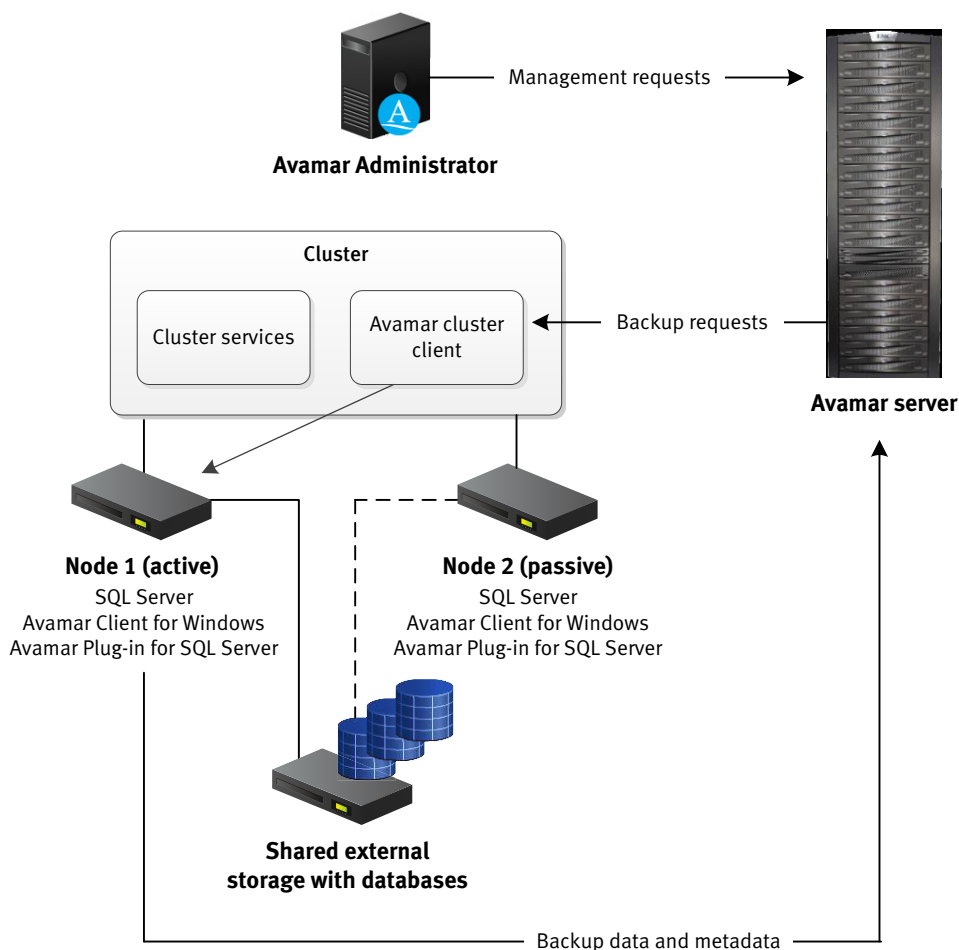


Figure 2 Avamar architecture with SQL Server in an active/passive cluster

You install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on each node in the cluster. Then you run the Cluster Configuration Tool to configure the Avamar cluster client.

The Avamar server sends backup requests to the Avamar cluster client, which passes the request to the SQL Server plug-in on the active node. The plug-in on the active node sends the backup data and metadata to the Avamar server.

Active/active cluster environment

In an active/active configuration, SQL Server runs on both nodes. The SQL Server installation on each node manages its own separate databases. If a failure occurs on one of the nodes, then the SQL Server installation on the other node assumes responsibility for managing the databases for both nodes. When the failed node comes back online, you can return the management responsibility for the databases to the original SQL Server installation on each node.

Avamar can perform both on-demand and scheduled backups of SQL Server data regardless of which active node is managing the data at the time of the backup.

The following figure illustrates an active/active cluster environment with SQL Server and Avamar.

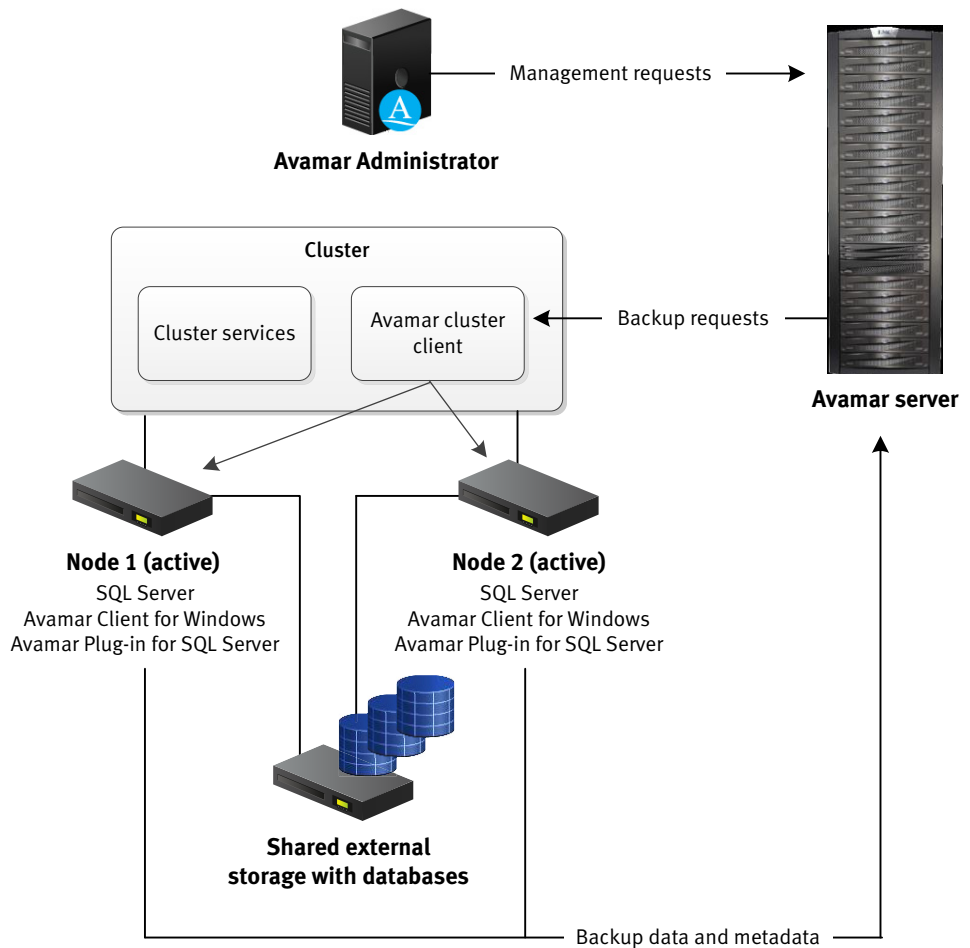


Figure 3 Avamar architecture with SQL Server in an active/active cluster

You install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on each node in the cluster. Then you run the Cluster ConfigurationTool to configure the Avamar cluster client. When you perform a backup or restore, you select the cluster client as the client to back up or the target client for the restore.

The Avamar server sends backup requests to the Avamar cluster client, which passes the request to the SQL Server plug-in on both active nodes. The plug-ins on the active nodes back up the SQL Server data that they are managing to the Avamar server.

AlwaysOn availability groups

In SQL Server 2012, you can configure high availability of groups of databases by using AlwaysOn availability groups (AGs). With AGs, SQL Server is installed in a WSFC cluster, but the data is not stored on a shared drive. Instead, the data is stored on each node, and SQL Server synchronizes the data from the primary version of the database to any secondary versions on other nodes.

You can group user databases together in an availability group. Databases in an availability group must use the full recovery model. All databases in an availability group fail over together from one node to another.

A SQL Server instance on a cluster node that hosts an availability group is called an *availability replica*. Each availability replica of an availability group must reside on a different node of the same cluster. There are two types of availability replicas: one primary replica and one to four secondary replicas. The primary replica handles read/write activity from clients and sends transaction log updates to the secondary replicas. Each secondary replica applies the transaction log updates to its databases. A maximum of two of the secondary replicas may be active at any time. The other two must be passive.

During failover of an AG, the target secondary replica assumes the primary role and becomes the new primary replica. The target secondary replica brings its databases online as the primary databases, and client applications can connect to them. When the former primary replica is available, it assumes the secondary role and becomes a secondary replica.

Architecture and workflow of backups in AlwaysOn availability groups

Avamar can perform both on-demand and scheduled backups of databases in either the primary replica or a secondary replica for an availability group.

You install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on each node in the cluster. Then you run the Cluster ConfigurationTool to configure the Avamar cluster client for the availability group listener. When you perform a backup or restore, you select the cluster client for the availability group listener as the client to back up or the target client for the restore.

[“Backups in AlwaysOn availability groups” on page 29](#) provides additional details on performing backups of databases in AlwaysOn availability groups.

Backups on the primary replica

The following figure illustrates the data workflow of SQL Server plug-in backups on the primary replica.

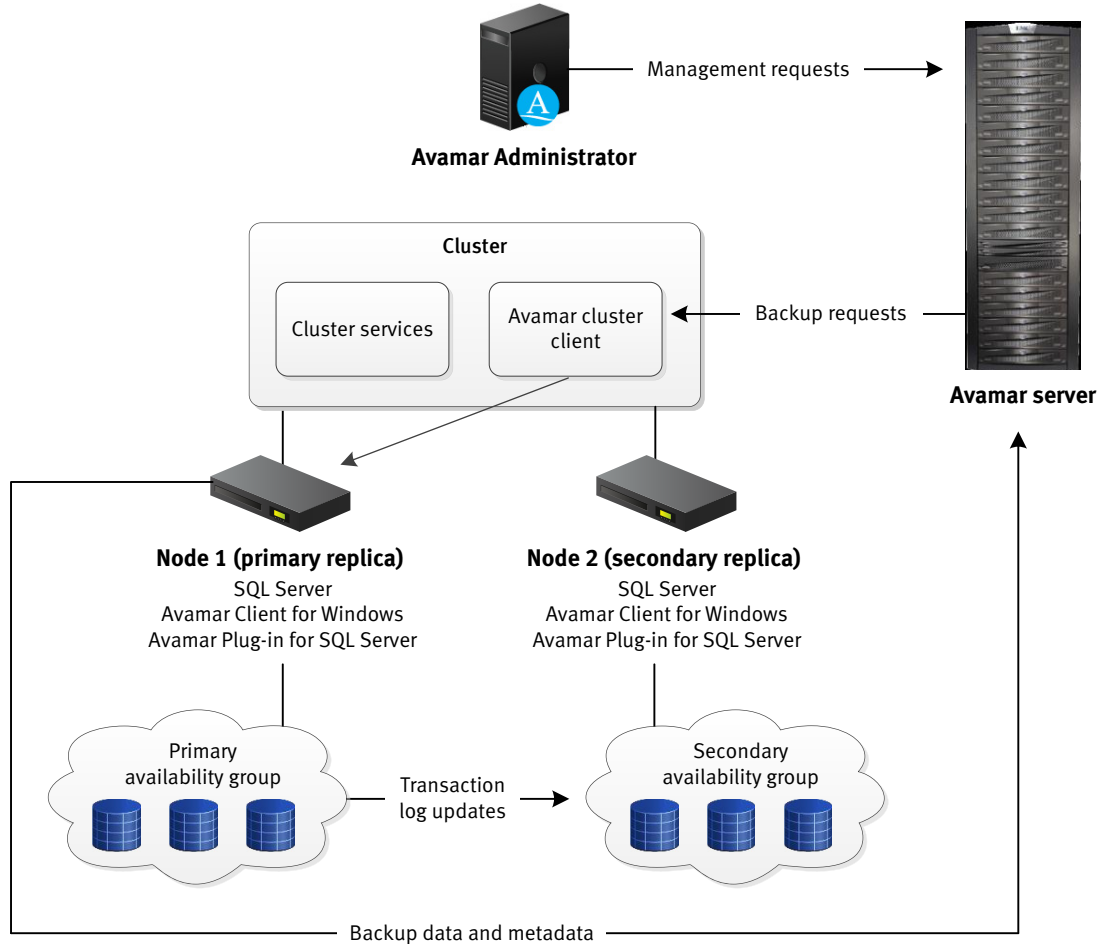


Figure 4 Avamar workflow for backup on a primary replica in an AlwaysOn environment

The Avamar server sends a backup request to the Avamar cluster client for the availability group listener, which passes the request to the primary replica. The SQL Server plug-in on the node with the primary replica performs the backup and sends the backup data and metadata to the Avamar server.

Backups on the secondary replica

The following figure illustrates the data workflow of SQL Server plug-in backups on a secondary replica.

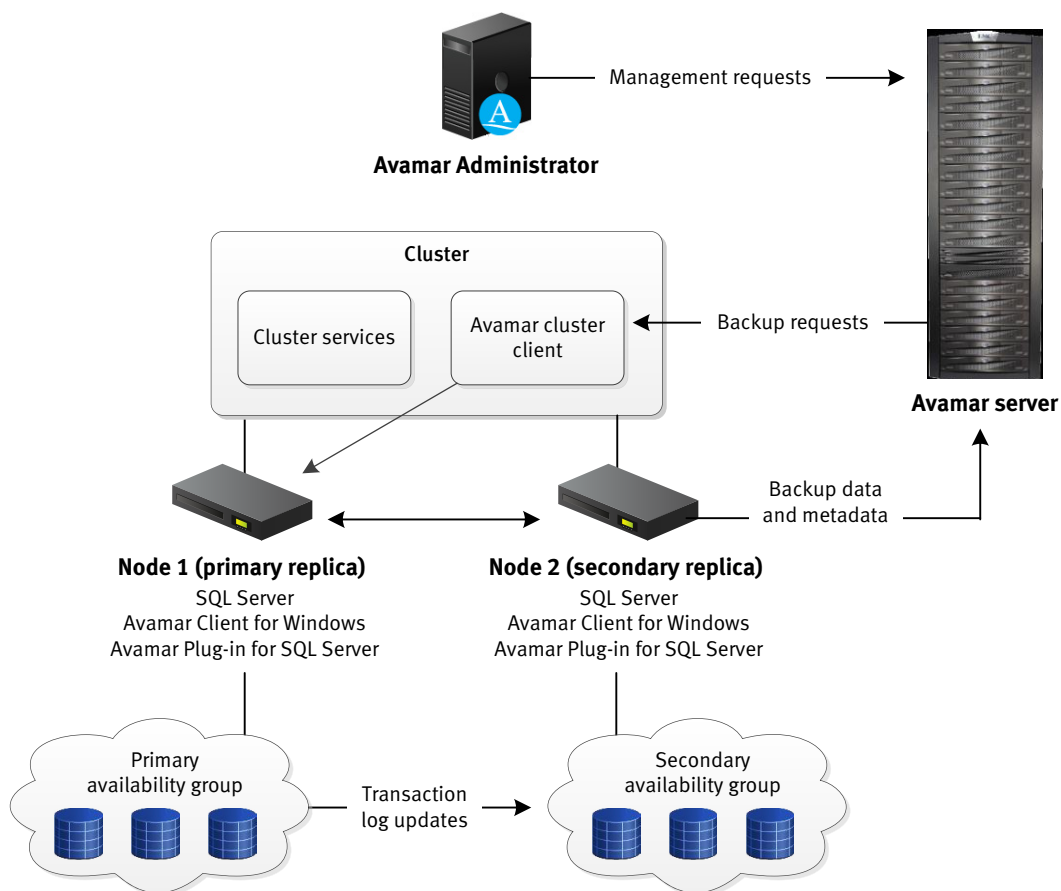


Figure 5 Avamar workflow for backup on a secondary replica in an AlwaysOn environment

The Avamar server sends a backup request to the Avamar cluster client for the availability group listener, which passes the request to the primary replica. The SQL Server plug-in on the node with the primary replica determines the secondary replica on which to perform the backup, and forwards the backup request to the SQL Server plug-in on the node with the secondary replica. The plug-in on the node with the secondary replica performs the backup and then sends the backup data to the Avamar server. The plug-in on the secondary replica then sends the metadata to the Avamar server and notifies the plug-in on the primary replica whether the backup completed successfully. The primary replica sends the final summary to the Avamar server.

Architecture and workflow of restores in AlwaysOn availability groups

You can use the backup from either the primary replica or a secondary replica to restore the databases in an AlwaysOn availability group. You can restore to only the primary replica, or restore to both the primary and secondary replicas.

[“Restore to an AlwaysOn availability group” on page 33](#) provides more information on restoring databases in AlwaysOn availability groups.

Restore to only the primary replica

The following figure illustrates the process to restore to only the primary replica in an AlwaysOn availability group.

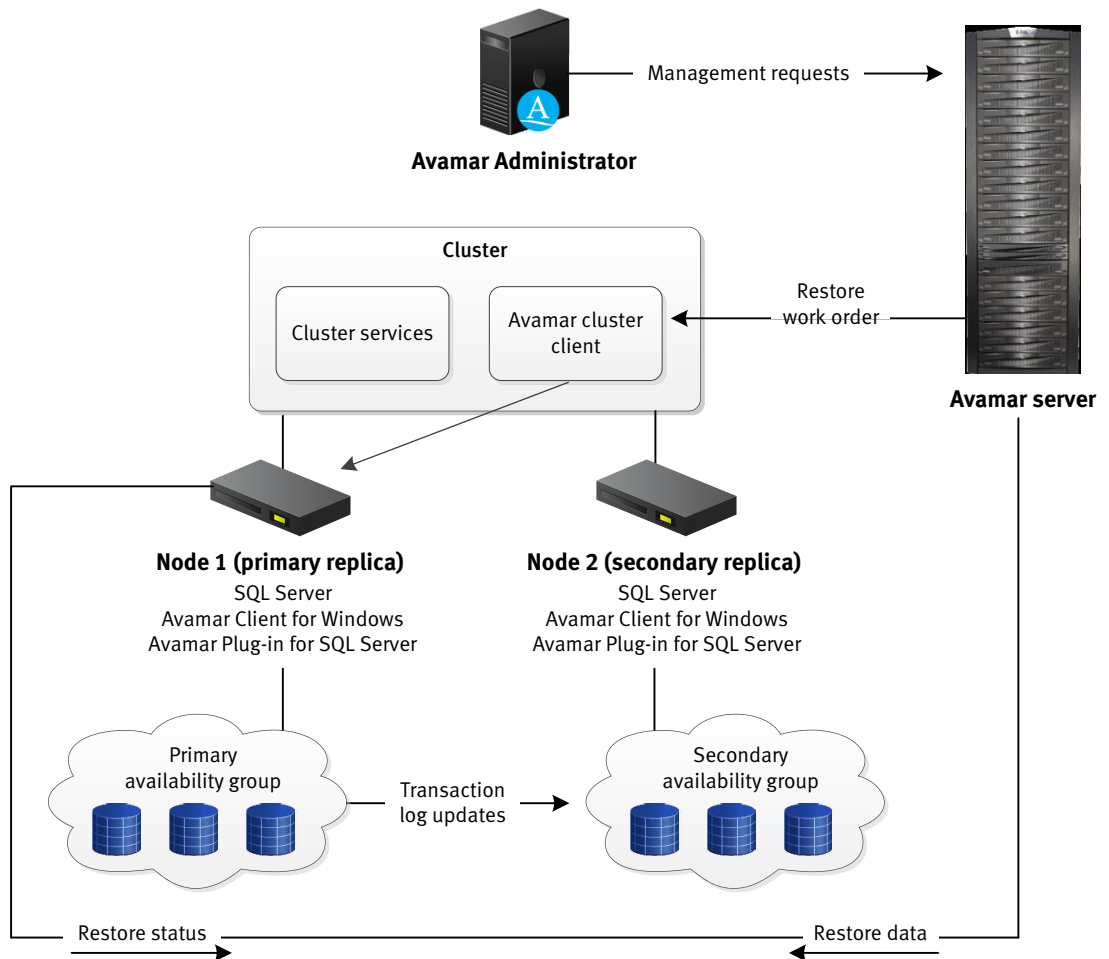


Figure 6 Avamar workflow for restore to only the primary replica in an AlwaysOn availability group

The Avamar server sends a restore work order to the cluster client for the availability group listener, which passes the work order to the SQL Server plug-in on the node with the primary replica. The plug-in on the primary replica removes the databases that you are restoring from the availability group and restores the databases. After the restore completes, the plug-in on the primary replica adds the restored databases to the availability group and sends information about the restore to the Avamar server.

If there is a corresponding database on the secondary replicas when you restore a database only on the primary replica, then the database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replica.

You can also set the database on a secondary replica online without rejoining it to the availability group by restoring the database with the `RECOVERY` recovery operation (`RESTORE database WITH RECOVERY`). The SQL Server 2012 documentation available on the Microsoft website provides details.

Restore to both the primary and secondary replicas

The following figure illustrates the process to restore to both the primary and secondary replicas.

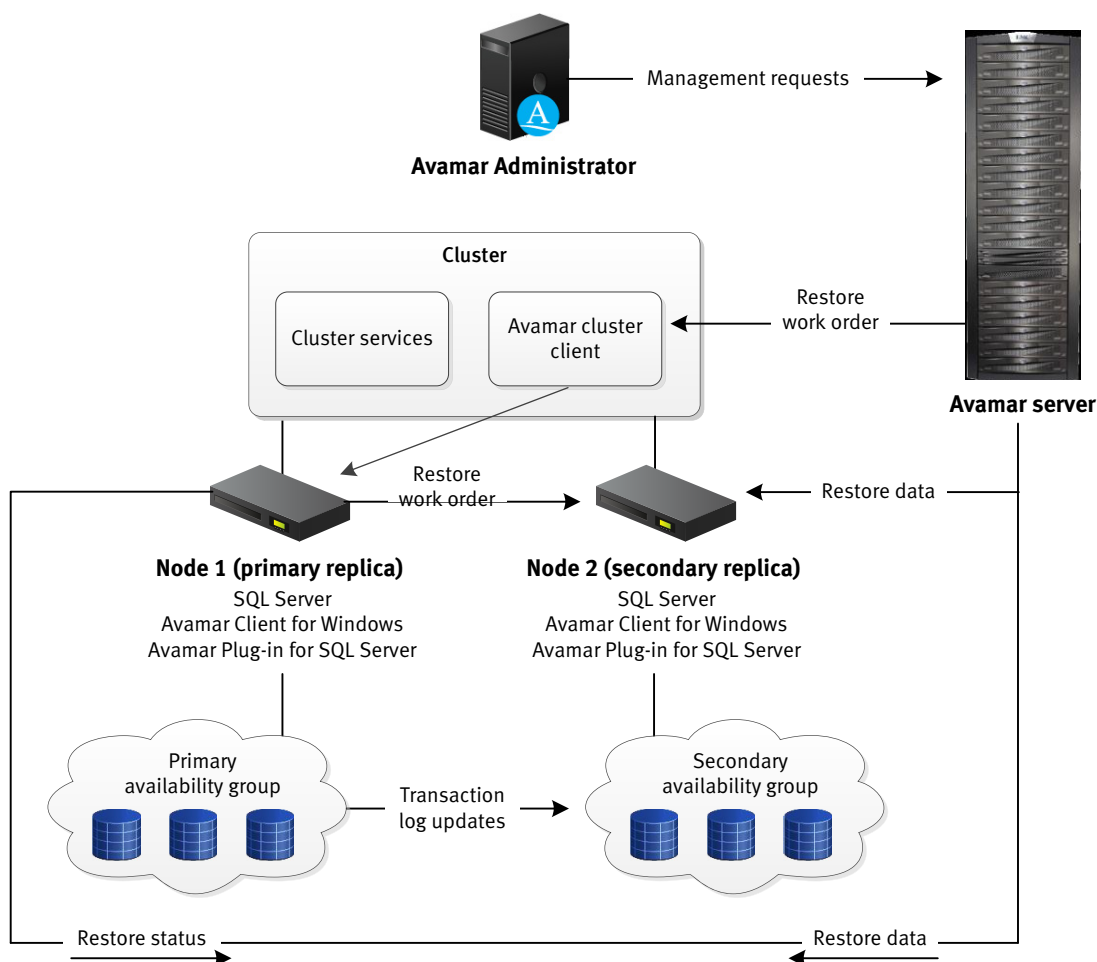


Figure 7 Avamar workflow for restore to the primary and secondary replicas in an AlwaysOn availability group

During the restore, the Avamar server sends a restore work order to the SQL Server plug-in on the node with the primary replica. The plug-in on the node with the primary replica then sends the restore work order to the plug-in on the nodes with the secondary replicas. The plug-ins on the primary and secondary replicas remove the databases that you are

restoring from the availability group and restore the databases. After the restore completes, the plug-in on the node with the primary replica adds the restored databases to the availability group. Then the plug-ins on the nodes with the secondary replicas join the restored databases to the availability groups. When the entire restore process completes on all replicas, the plug-in on the node with the primary replica sends information about the restore to the Avamar server.

Database mirroring

Database mirroring maintains multiple copies of a single database that must reside on different SQL Server instances. Typically, these server instances reside on computers in different locations.

The principal server serves the database to clients, while the mirror servers serve as standby servers.

You implement mirroring settings individually for each database.

Mirroring works only with databases that use the full recovery model. The simple and bulk-logged recovery models do not support database mirroring, and you cannot mirror the master, msdb, tempdb, or model databases.

You can use the Avamar Plug-in for SQL Server to back up mirrored SQL Server databases. However, several conditions apply. [“Database mirroring requirements” on page 44](#) provides details.

Data Domain system support

You can store backups on either the Avamar server or an EMC Data Domain[®] system. Backup metadata is stored on the Avamar server.

Before you can store backups on a Data Domain system, you must add the Data Domain system to the Avamar configuration by using Avamar Administrator. Then you select the Data Domain system in the plug-in options when you perform an on-demand backup or when you create a dataset for a scheduled backup. You can also use the CLI to perform backups to a Data Domain system.

You must store the full backup for a client and all subsequent transaction log (incremental) and differential backups on either the Avamar server or a single Data Domain system. [“Mixed backups” on page 30](#) provides additional details.

The steps to restore backups are the same whether you restore from the Avamar server or a Data Domain system. The restore process determines the location of the backup and restores the backup.

The *EMC Avamar and EMC Data Domain System Integration Guide* provides more information about Data Domain systems in an Avamar environment, including detailed steps to add a Data Domain system to the Avamar configuration.

Server virtualization

You can install SQL Server in a server virtualization environment such as VMware or Microsoft Hyper-V. There are multiple ways that you can install and use Avamar to back up and restore SQL Server data in a server virtualization environment. The *EMC Avamar for Hyper-V VSS User Guide* and *EMC Avamar for VMware User Guide* provide details on additional system requirements, as well as installation and configuration procedures.

Backup

When you perform a backup with the Avamar Plug-in for SQL Server, you can back up either all SQL Server data on a specific server, one or more instances, or one or more databases. You cannot use the Avamar Plug-in for SQL Server to back up individual filegroups, files, or tables. To back up individual files, use the Avamar Client for Windows.

Avamar can back up databases that use any of the three recovery models: simple, full, or bulk-logged. However, the recovery model may determine the type of backup that you can perform of the database. [“Choosing a backup type based on the recovery model” on page 28](#) provides details.

If the databases are in AlwaysOn availability groups, then you can select the replica on which the backup occurs. [“Backups in AlwaysOn availability groups” on page 29](#) provides details.

You can store backups on either the Avamar server or a Data Domain system. However, mixed backups are not supported. [“Mixed backups” on page 30](#) provides details.

You also can improve backup performance with multi-streaming. [“Multi-streaming” on page 30](#) provides more information.

Backup types

The Avamar Plug-in for SQL Server supports three types of database backups:

- ◆ Full
- ◆ Differential
- ◆ Transaction log (incremental)

The following figure illustrates the different types of data that are included in each of the backup types.

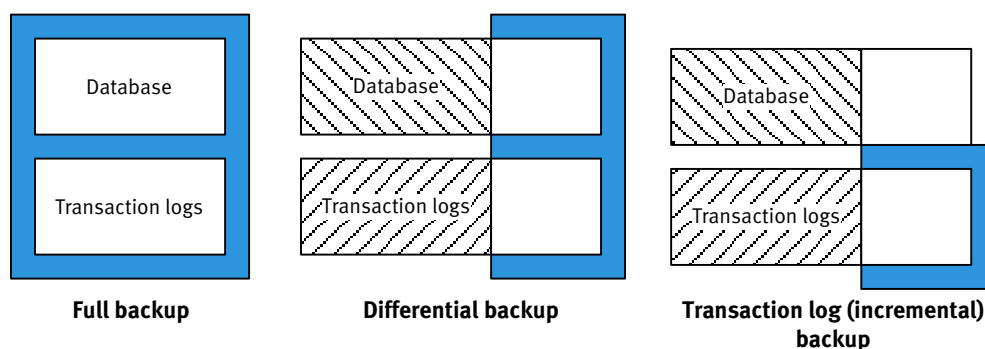


Figure 8 Types of database backups

Full backups

Full backups include the entire database, including all objects, system tables, and data. As the backup operation progresses, it copies the transaction logs. The timing of the transaction log backup ensures that you can recover the complete database to the state it was in when the backup finished.

Differential backups

Differential backups include data that changed after the last full backup. The backup operation also copies relevant portions of the transaction logs.

When you select a differential backup to restore a database, the restore process restores the last full backup, followed by the differential backups performed after the full backup. Because a differential backup only saves changes to data, the backup is smaller and faster than a full backup. Therefore, you can perform a differential backup more often than a full backup.

Differential backups are used primarily in heavily utilized systems where you must bring a failed database back online as quickly as possible. Differential backups are smaller than full backups, and so have less effect on the system where they run.

Transaction log (incremental) backups

By default, transaction log backups only back up the transaction logs. Transaction logs are serial records of all database modifications. The logs are used in recovery operations to update the database with complete transactions and roll back incomplete transactions. When you back up a transaction log, the backup stores all changes after the last transaction log backup. Transaction log backups record the state of the transaction log at the start of the backup (unlike full and differential backups, which record the state of the data at the end of the backup).

When a transaction log backup is complete, the log truncates to remove any transactions committed to the database. When restoring the database, you restore the data to the state it was in at the end of the last full or differential backup, and then sequentially restore the transaction log backups in order.

To fully restore data from a transaction log backup, at least one full backup must exist. Therefore, to ensure data integrity, the Avamar Plug-in for SQL Server software always checks for the presence of a full backup on the server. If there is a full backup, the transaction log backup proceeds (that is, the backup includes only transaction logs). If there is no full backup and you select the **Force full backup** option (the default setting), then the Avamar Plug-in for SQL Server software forces a full backup to ensure data integrity.

Choosing a backup type based on the recovery model

You can perform full backups of all databases, regardless of the recovery model (simple, full, or bulk-logged).

If the database uses the simple recovery model, then you cannot perform a transaction log backup of the database. Databases with the simple recovery model do not support transaction log backups. Databases that use the simple recovery model include system databases, such as the master and msdb databases.

Microsoft SQL Server also does not allow differential backups of the master database. You can only create full backups of the master database.

If a transaction log backup includes databases with the simple recovery model and databases with other recovery models, then you can select how Avamar handles the databases with the simple recovery model. You can exclude the databases with the simple

recovery model and log either a warning or an error message in the log, or you can automatically perform a full backup of the databases. When you perform the backup, the **For simple recovery model databases** option controls this behavior.

Databases in AlwaysOn availability groups must use the full recovery model.

If you change the recovery model of a database, perform a full backup before you try to perform a differential or transaction log backup.

Backups in AlwaysOn availability groups

Avamar can perform both on-demand and scheduled backups of databases in either the primary replica or a secondary replica for an availability group. You can increase performance of the primary replica when you perform backups on a secondary replica.

Settings in SQL Server and in Avamar enable you to specify a preference for the replica on which the backup occurs:

- ◆ Always on the primary replica
- ◆ Always on a secondary replica
- ◆ On a secondary replica if possible, otherwise on the primary replica
- ◆ No preference defined

If you do not specify a preference, then other factors contribute to the selection of the replica for backup, including the backup priority, operational status, and connection status of each replica. [“AlwaysOn availability group option” on page 82](#) provides details on setting the preference for the replica.

If you perform the backup on a secondary replica, then the replica must meet the following requirements:

- ◆ The replica must be in a synchronized state.
- ◆ The replica role setting in SQL Server must allow either read-only connections or all connections.
- ◆ If there are multiple secondary replicas, then set the priority for which secondary replica to use for the backup.

[“AlwaysOn availability group requirements” on page 43](#) provides additional details on system requirements. If you select the option to perform the backup only on a secondary replica and no secondary replicas that meet the requirements are available, then the backup fails with an error.

You can perform full, differential, and transaction log backups on the primary replica. You can perform only full and transaction log backups on a secondary replica. If you try to perform a differential backup on a secondary replica, then the backup fails with an error message.

SQL Server does not support system databases in an availability group. To back up system databases with Avamar, you must perform a separate backup. During the backup, select the server name of the physical node as the client to back up, and then select the system databases.

Mixed backups

You can store backups on either the Avamar server or a Data Domain system. However, the full backup for a client and all subsequent transaction log (incremental) and differential backups must be stored on either the Avamar server or a single Data Domain system. Avamar does not support the following scenarios:

- ◆ Full backup on a Data Domain system, and transaction log or differential backups on the Avamar server
- ◆ Full backup on the Avamar server, and transaction log or differential backups on a Data Domain system
- ◆ Full backup on one Data Domain system, and transaction log or differential backups on another Data Domain system

NOTICE

An exception to this restriction is if you perform a tail-log backup during a restore. The tail-log backup is stored on the Avamar server even if the other backups for the client are stored on a Data Domain system.

As a result, if you change the server on which backups for a client are stored, then you must perform a full backup before any further transaction log or differential backups.

If you change the backup storage from one Data Domain system to another Data Domain system, you can restore to a point in time between the two full backups. [“Requirements for point-in-time restore after changing the backup storage” on page 97](#) provides more information.

Multi-streaming

Multi-streaming enables you to improve backup and restore performance by backing up and restoring SQL Server data using multiple parallel data streams. You can either back up multiple databases in parallel with one stream per database, or back up a single database using multiple parallel streams.

If you use multiple data streams to send backup data for a single database to the Avamar server or Data Domain system, then the backup for the database is stored as multiple files. As a result, the restore uses the same number of streams that you use for the backup.

You can specify a maximum of 10 streams for each backup. You also can specify the minimum size of a stream.

[“Multi-streaming options” on page 85](#) provides details on how multi-streaming works and how to set multi-streaming options to maximize performance.

Restore

The following restore scenarios are available when you use the Avamar Plug-in for SQL Server to back up a SQL Server environment:

- ◆ Restore an instance, database, filegroup, or file to its original location.
- ◆ Restore a database to its original instance, but with a new name.
- ◆ Restore an instance, database, filegroup, or file to a different instance on the original server.
- ◆ Restore an instance, database, filegroup, or file to an instance on a different server.
- ◆ Restore an instance or database to operating system files.
- ◆ Restoring a database in an AlwaysOn availability group.
- ◆ Restore a lost or corrupt database with an intact database log file.
- ◆ Restore system databases to the original location.

The SQL Server plug-in performs only offline restores of SQL Server data. Online restore is not supported.

Avamar provides several options to control restore behavior, including tail-log backup, point-in-time restore, and SQL Server recovery operations.

Restore to the original location

There are two options when you use the Avamar Plug-in for SQL Server to restore a SQL Server instance, database, or filegroup to its original location:

- ◆ Perform a standard restore with a tail-log backup and recovery.
- ◆ Use the SQL Server `REPLACE` option to completely overwrite the database.

A standard restore with a tail-log backup is perhaps the most common restore procedure. During this procedure, a tail-log backup is created to capture transactions that are not in a backup. Then Avamar restores the database from the most recent full backup and any differential or transaction log backups.

You may need to use the SQL Server `REPLACE` option for a restore, for example, if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log:

```
One or more devices or files already exist.
Reissue the statement using the WITH REPLACE
option to overwrite these files and devices.
```

NOTICE

When you use the Avamar plug-in option for the SQL Server `REPLACE` option, it adds an SQL `WITH REPLACE` clause statement to the restore Transact-SQL command. Use of this statement overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The *Microsoft Transact-SQL Reference Manual* provides more information about the safety check in the `RESTORE` command section.

[“Restoring to the original location” on page 106](#) provides more information.

Restore to a new database in the original instance

You can use the Avamar Plug-in for SQL Server to restore a backup of a database to its original instance but with a new name, creating a new database in the instance. [“Restoring to a new database in the original instance” on page 109](#) provides instructions.

When you restore to a new database in the original instance, you can perform a tail-log backup and recovery to recover transactions that were not in the backup. [“Tail-log backup” on page 36](#) provides additional details and requirements for tail-log backups.

You also can specify the path for the database and log files.

If the database uses the full recovery model, then you can restore to either a specific date and time or to a named mark in the transaction log. [“Point-in-time restore” on page 37](#) provides additional details on point-in-time restore.

Restore to a different instance on the original server

You can use the Avamar Plug-in for SQL Server to restore an instance, database, filegroup, or file to a different instance on the original server. [“Restoring to a different instance” on page 113](#) provides instructions.

When you restore to a different instance on the original server, you cannot perform a tail-log backup. However, you can specify the path for the database and log files.

If you are restoring a database, you can leave the original database name or restore the database with a new name. You can also restore to either a specific date and time or to a named mark in the transaction log. [“Point-in-time restore” on page 37](#) provides additional details on point-in-time restore.

Restore to an instance on a different server

You can use the Avamar Plug-in for SQL Server to restore an instance, database, filegroup, or file to an instance on a different server. [“Restoring to a different instance” on page 113](#) provides instructions.

When you restore to an instance on a different server, you cannot perform a tail-log backup. However, you can specify the path for the database and log files.

If you are restoring a database, you can leave the original database name or restore the database with a new name. You also can restore to either a specific date and time or to a named mark in the transaction log. [“Point-in-time restore” on page 37](#) provides additional details on point-in-time restore.

Restore to a file

If the Avamar Plug-in for SQL Server is not installed on the target server, or you want to use the standard SQL Server restore tools for other features, then you can restore a database to operating system files. You can then use SQL Server tools, such as SQL Server Management Studio, a Transact-SQL `RESTORE` command, or the Microsoft SQL Server `sqlcmd` utility, to restore the database.

The SQL Server plug-in restores the backup as one or more files to the specified destination in the following path:

```
destination\instance\database\file
```

where:

- ◆ *destination* is the destination for the files that you specified in the Set Destination dialog box.
- ◆ *instance* is the name of the SQL Server instance from the backup.
- ◆ *database* is the name of the database from the backup.
- ◆ *file* is the name of the file. A single backup may include multiple files, depending on the number of streams in the backup. The file name for each file is composed of the backup type and the stream number:
 - f-0 indicates a full backup
 - d-*n* indicates a differential backup
 - i-*n* indicates a transaction log (incremental) backup

where *n* is the sequential number of the differential or incremental backup after the preceding full backup.

For example, if there are two files, f-0.stream0 and f-0.stream1, then the backup files are for a full backup with two streams.

[“Restoring to a file” on page 119](#) provides more information.

Restore to an AlwaysOn availability group

When you back up a database in an AlwaysOn availability group, you can restore the database to the following locations:

- ◆ To the original availability group
- ◆ To a new database in the original availability group
- ◆ To a different availability group
- ◆ To a different instance that does not use availability groups
- ◆ To a file

You can restore an entire database, a filegroup, or a file. You also can perform point-in-time restore of databases that use the full recovery model.

These restore options are available regardless of whether the backup was on the primary replica or a secondary replica.

Restore to the original availability group

When you restore to the original availability group, the restore process can automatically restore the databases on both the primary replica and the secondary replicas.

You can also restore databases only on the primary replica. When you restore a database only on the primary replica, the corresponding database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replicas.

You can also set the databases on a secondary replica online without rejoining them to the availability group by restoring the databases with the `RECOVERY` recovery operation. The SQL Server 2012 documentation available on the Microsoft website provides details.

[“Restoring to the original availability group” on page 138](#) provides instructions.

Restore to a new database in the original availability group

You can restore a database to the original AlwaysOn availability group but with a new name, creating a new database in the availability group. When you restore to a new database in the original instance, the restore process restores the database to only the primary replica and you must join the new database to the availability group.

To restore the database on the secondary replicas as part of the availability group, manually restore the database and join the database to the availability group on each secondary replica.

[“Restoring to a new database in the original availability group” on page 139](#) provides instructions.

Restore to a different availability group

When you restore a database to an availability group on a different server, the restore process restores the database to only the primary replica.

To restore the database on the secondary replicas as part of the availability group, manually restore the database and join the database to the availability group on each secondary replica.

Note: You cannot restore a database to a different availability group in the same cluster. Restore fails with an error message that the database already exists.

[“Restoring to a different availability group” on page 140](#) provides instructions.

Restore to a different instance that does not use availability groups

You can restore a database to a different SQL Server instance that does not use AlwaysOn availability groups.

The steps are the same whether you restore from a backup of databases on a stand-alone server, in a cluster, or in an availability group. [“Restoring to a different instance” on page 113](#) provides instructions.

Restore of a database from an availability group to a file

You can restore a database from a backup of an availability group to operating system files. This type of restore may be necessary if the Avamar Plug-in for SQL Server is not installed on the target server, or you want to use the standard SQL Server restore tools for other features.

The steps are the same whether you restore from a backup of databases on a stand-alone server, in a cluster, or in an availability group. [“Restoring to a file” on page 119](#) provides instructions.

Restore of a database with an intact log file

If a database becomes corrupt or is otherwise lost but an intact database log file is available, you can restore the database and use the log file to recover transactions after the most recent Avamar backup.

To restore the database in this scenario, you perform a transaction log backup using a Transact-SQL command. Then, you restore the database from the most recent Avamar backup using Avamar Administrator. Finally, you restore the transaction log backup using Transact-SQL commands. [“Restoring a database with an intact log file” on page 141](#) provides instructions.

Restore of system databases

It is rare that you need to restore only system databases. However, the restore may be necessary if one or more system databases are damaged.

It is more likely that you must restore system databases when you restore user databases. When you select both system and user databases for restore, the system databases restore first.

When you restore system databases, the Avamar Plug-in for SQL Server can automatically restore the databases in the correct order and manage SQL Server services. However, you can also restore individual system databases and manually manage the services.

Automatic restore of system databases

When you restore multiple system databases, Avamar automatically restores the databases in the correct order: master, msdb, and model.

Avamar can also automatically manage the stop and restart of the necessary SQL Server services during the restore:

- ◆ When you restore the master database, Avamar can automatically stop the SQL Server instance, including dependent services such as the SQL Server agent service and the Analysis Service, and restart the instance in single-user mode before the restore. After the restore, Avamar automatically restarts the instance.
- ◆ When you restore the msdb database, Avamar can automatically stop the SQL Server agent service, and then restart it when the restore is complete.

[“Restoring system databases automatically” on page 132](#) provides instructions.

Manual restore of system databases

When you restore system databases manually, you must perform the following high-level steps:

1. Shut down the SQL Server instance and then restart the instance in single-user mode.
2. Restore the master database.
3. Restart the SQL Server service.
4. Stop the SQL Server Agent service.
5. Restore the msdb and model databases.
6. Restart the SQL Server Agent service.

[“Restoring system databases manually” on page 135](#) provides instructions.

Tail-log backup

With a tail-log backup, Avamar backs up the tail of the transaction log during the restore process to capture the log records that are not in a backup. After the database restore, Avamar uses the tail-log backup to recover the transactions that were not in the backup.

To perform a tail-log backup, the database must be online and using either the full or bulk-logged recovery model. As a result, you cannot perform a tail-log backup of system databases such as the master and msdb databases because those databases use the simple recovery model.

When you are restoring a user-defined filegroup or secondary data file and you perform a tail-log backup, you must select the most recent backup as the backup from which to restore. Otherwise, the restore fails and an error message is written to the log file.

You can perform a tail-log backup when you are restoring an instance, database, filegroup, or file to its original location *without* the SQL WITH REPLACE option.

You can also perform a tail-log backup when you are restoring a database to the original instance but with a new database name.

If you are performing a point-in-time restore and the point in time that you are restoring to is after the most recent transaction log backup, then you *must* perform a tail-log backup.

A tail-log backup is also necessary if you restore a file from a user-defined filegroup to its original location.

Do *not* perform a tail-log backup if you are performing a redirected restore to a different SQL Server instance.

NOTICE

If the tail-log backup fails to complete, then the restore cannot take place. Review the log file to determine the cause of the failure. Correct the problem, and then restart the restore. Keep in mind that if you clear the **Tail-log backup** checkbox to prevent the tail-log backup from occurring, then the restore includes only the transactions up to the selected backup. Any transactions in the tail of the log are lost.

Point-in-time restore

You can restore a database with the full recovery model to a specific date and time or to a named mark in the transaction log.

You cannot perform a point-in-time restore of system databases such as the master and msdb databases because those databases use the simple recovery model.

To restore to a specific point in time, you must provide the transaction date and time or named mark to which to recover. This information is available in the SQL Server transaction log. The SQL Server documentation on the Microsoft website provides details on how to access transaction log information.

The point in time to which you are restoring must be after the finish time for the most recent full backup. In addition, if the point in time is before the start time of the most recent transaction log (incremental) backup, then a tail-log backup is not necessary. However, a tail-log backup *is* required if the point in time is after the most recent transaction log backup.

When you specify the point in time for restore, do not specify the start time of the selected transaction log backup if it is not the last backup in the backup sequence. Otherwise, the restore fails, and a tail-log backup does not occur even if you select the **Tail-log backup** checkbox.

SQL Server recovery operations

You can control the recovery operation that occurs after the restore using restore options in the Avamar Plug-in for SQL Server.

Table 2 Recovery operation options

Recovery operation	Description
RECOVERY	The database is fully recovered and online after the restore. This operation is the default setting.
NORECOVERY	The database remains in a restoring state after the restore. This enables you to perform additional manual restore tasks, such as applying additional SQL transaction log files.
STANDBY	The database is in standby (read-only) mode after the restore. This mode enables you to bring up a database for read-only access between transaction log restores. You may need to use this option with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores. This option also creates a file with recovery changes. You can use the file to revert the recovery changes, if required.

[“Recovery operation options” on page 145](#) provides details on selecting recovery operations in different restore scenarios.

Disaster recovery

To ensure sufficient preparation for disaster recovery of a SQL Server environment, you must perform ongoing backups of the Windows server and all system and user databases. [Chapter 5, “Disaster Recovery,”](#) provides high-level procedures for preparing for and performing disaster recovery.

CHAPTER 2

Installation

The following topics describe how to install, upgrade, and uninstall the Avamar Plug-in for SQL Server:

- ◆ [Preparing to install the SQL Server plug-in.....](#) 40
- ◆ [Installing the Avamar client software.....](#) 49
- ◆ [Upgrading the Avamar client software](#) 56
- ◆ [Uninstalling the Avamar client software.....](#) 58

Preparing to install the SQL Server plug-in

Before you install the Avamar Plug-in for SQL Server, verify the system requirements and prepare the environment.

Checking the system requirements

Before you install Avamar software on a client system, ensure that the client operating system and hardware configuration are supported.

The Avamar file system client and the SQL Server plug-in that you install on the host must be the same version number.

Up-to-date client compatibility information is available in the *EMC Avamar Compatibility and Interoperability Matrix* on EMC Online Support at <https://support.EMC.com>. The requirements in the matrix include supported operating systems and application versions.

Installation of SQL Server 2008 or SQL Server 2008 R2 in a Windows Server 2012 cluster requires SQL Server 2008 SP3 or later, or SQL Server 2008 R2 SP1. “[Preparing a cluster](#)” on [page 46](#) provides instructions for installing the service pack version of SQL Server.

On Windows Server 2012, the Avamar Plug-in for SQL Server does not support backup and restore of SQL Server data on Server Message Block (SMB) 3.0 file shares.

The environment must also meet other requirements, including hardware requirements, security requirements, and SQL Server feature and configuration requirements.

Hardware requirements

The following table lists the hardware requirements for the Avamar Plug-in for SQL Server.

Table 3 Minimum and recommended hardware requirements

Requirement	Minimum and recommended
RAM	512 MB (2 GB recommended).
Hard drive space	2 GB permanent hard drive space for software installation. The Avamar Plug-in for SQL Server software also requires an additional 12 MB of permanent hard drive space for each 64 MB of physical RAM. This space is necessary for local cache files.
Network interface	10BaseT minimum; 100BaseT or higher recommended, configured with the latest drivers for the environment.

Security requirements

When Avamar connects to SQL Server for backup or restore, you can use either Windows (NT) authentication or SQL Server authentication:

- ◆ With Windows authentication, Avamar connects to SQL Server by using the Windows system service account (NT AUTHORITY\SYSTEM), which has privileges in SQL Server. The account must have the sysadmin server-level role.

Note: To use the Windows authentication method, select **NT authentication** in the plug-in options for the Avamar Plug-in for SQL Server.

- ◆ With SQL Server authentication, Avamar connects to SQL Server using a SQL Server login account. The account must have the sysadmin server-level role. You must select the mixed authentication mode for the Database Engine when you configure SQL Server.

If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the Windows system service account. This account has the public and sysadmin server-level roles in SQL Server by default in SQL Server 2008, 2008 R2, and 2005. In SQL Server 2012, you must add the account to the SQL Server administrators group.

Adding the Windows system service account as an administrator

1. In SQL Server Management Studio, expand the **Security** node and then the **Logins** node for the instance in the left pane.
2. Right-click the **NT AUTHORITY\SYSTEM** account and select **Properties**.
The **Login Properties** dialog box appears.
3. Select the **Server Roles** page from the list in the left pane.
4. In the right pane, select the checkbox next to the **sysadmin** user.
5. Click **OK**.

Avamar server dependencies

The following table lists Avamar server and SQL Server plug-in version compatibility requirements.

Table 4 Avamar server and SQL Server plug-in compatibility

Avamar server version	Supported SQL Server plug-in versions
7.0	7.0, 6.1
6.1	7.0, 6.1
6.0	6.0, 5.0
5.0	6.0, 5.0

The user interface and underlying code for several features, including point-in-time restore and redirected restore, changed in release 6.1. The changes require release 6.1 or 7.0 on both the client and server.

If you must upgrade the Avamar server to release 6.1 or 7.0 but you cannot upgrade the SQL Server plug-in, then use the release 6.0 plug-in catalog on the Avamar server. Contact EMC Customer Support for assistance with using a different plug-in catalog.

Checking the SQL Server requirements

The following topics list SQL Server requirements for the Avamar Plug-in for SQL Server.

SQL Server feature installation requirements

The Avamar Plug-in for SQL Server requires installation of SQL Server Management Objects (SMO). To install SMO, install the Client Tools SDK when you install SQL Server. Alternatively, install SMO from the SQL Server feature pack.

You also must install Management Tools and the SQL Client Connectivity SDK when you install SQL Server.

Database name requirements

Use only valid file name characters in database names. Avoid any of the following characters, which are known to interfere with proper operation of the SQL Server plug-in: asterisk (*), forward slash (/), backward slash (\), colon (:), semicolon (;), question mark (?), right angle bracket (>), left angle bracket (<), vertical bar (|), or number (#).

Also, do not end the database name with a period (.). If you end a database name with a period and then you restore the database, you receive an access violation when you try to open the folder for the restored database.

Transaction log marking requirements

The Avamar Plug-in for SQL Server cannot successfully back up and restore databases in which database transaction log marks contain leading or trailing white space.

Resource database requirements

The SQL Server Resource database is a read-only database that contains copies of all system objects that ship with SQL Server. The Avamar Plug-in for SQL Server cannot back up the Resource database, `mssqlsystemresource.mdf`, or its log file, `mssqlsystemresource.ldf`, because SQL Server cannot back up the Resource database. To back up Resource database files, use the Avamar Client for Windows.

Alias requirements

A SQL Server alias is a name that you configure on the client computer that points to a server, instance, or database on either the local server or on a different computer. You can use an alias to connect with a certain network protocol, such as TCP/IP, Named Pipes, or Shared Memory.

An alias can improve the convenience and speed of connecting to SQL Server. However, improper use of an alias can result in connectivity issues that are difficult to isolate and troubleshoot.

If you plan to use a SQL Server alias, review the requirements and configuration steps on the Microsoft website for setting up an alias. Also, review best practices for using and troubleshooting aliases. These best practices are available in the blogs and Support knowledgebase articles on the Microsoft website.

AlwaysOn availability group requirements

You can back up databases on either the primary or secondary replica for an availability group.

There must be an availability group listener for each availability group.

If you perform the backup on a secondary replica, then the replica must meet the following requirements:

- ◆ The replica must be in a synchronized state.
- ◆ The replica role setting in SQL Server must allow either read-only connections or all connections:
 - When you create a replica in SQL Server Management Studio, select either **Read-intent only** or **Yes** from the **Readable secondary** list on the **Replicas** tab in the **Add Replica Wizard**.
 - When you edit the configuration for an availability group in SQL Server Management Studio, select either **Read-intent only** or **Yes** from the **Readable secondary** list on the **Availability Replica Properties** dialog box.
 - Specify either `READ_ONLY` or `ALL` for the `SECONDARY_ROLE` option when you issue the `CREATE AVAILABILITY GROUP` or `ALTER AVAILABILITY GROUP` Transact-SQL command.
- ◆ If there are multiple secondary replicas, then set the priority for which secondary replica to use for the backup:
 - When you create a replica in SQL Server Management Studio, specify a value between 1 and 100 in the **Backup Priority** field in the **Add Replica Wizard**.
 - When you edit the configuration for an availability group in SQL Server Management Studio, specify a value between 1 and 100 in the **Backup Priority** field on the **Availability Replica Properties** dialog box.
 - Specify a numeric value for the `BACKUP_PRIORITY` option when you issue the `CREATE AVAILABILITY GROUP` or `ALTER AVAILABILITY GROUP` Transact-SQL command.

A value of 1 indicates the lowest priority, and a value of 100 indicates the highest priority.

SQL Server does not support system databases in an availability group. To back up system databases with Avamar, you must perform a separate backup. During the backup, select the server name of the physical node as the client to back up, and then select the system databases.

Database mirroring requirements

You can use the Avamar Plug-in for SQL Server to back up SQL Server databases that are mirrored. However, the following conditions apply:

- ◆ The SQL Server version must be 2008, 2008 R2, or 2012.
- ◆ Perform backups of only the principal database, not the mirrors.
- ◆ Use only the Avamar Plug-in for SQL Server to perform backups. If you use other backup products in addition to the Avamar Plug-in for SQL Server, then log chain breaks may occur.
- ◆ Backup and database administrators must use extra care. Some mirror operations cause log chain breaks that cannot be detected.
- ◆ When database mirroring is established, either initially or as the result of failover and failback, then you must manually perform a new full backup. Otherwise, incremental and differential backups that occur after the establishment of database mirroring are not valid for restore. After you perform the full backup in this case, you cannot perform point-in-time recoveries to a point in time before the full backup.
- ◆ To restore a database, you must break the SQL mirror.

Transparent Data Encryption requirements

When you protect Avamar backups of SQL Server 2008, SQL Server 2008 R2, or SQL Server 2012 databases with Transparent Data Encryption (TDE), the backups intentionally do not include the Database Encryption Key (DEK).

Including the DEK defeats one of the primary reasons for using TDE, which is to encrypt exported data such as backups.

When you perform an Avamar backup of TDE-protected databases, manually back up the DEK. The article “Understanding Transparent Data Encryption (TDE),” available on the Microsoft TechNet website, provides additional details.

FILESTREAM requirements

The SQL Server FILESTREAM feature enables you to configure a database to store binary large object (BLOB) data as files on the file system instead of in the database. The Avamar Plug-in for SQL Server supports backups of SQL Server databases with FILESTREAM data for the following SQL Server versions:

- ◆ SQL Server 2012
- ◆ SQL Server 2008 R2
- ◆ SQL Server 2008

The steps to back up and restore a database with FILESTREAM data by using either Avamar Administrator or the `avsql` command line interface (CLI) are the same as the steps to back up and restore other databases.

Known issues with FILESTREAM support

You might encounter the following issues when you back up or restore a SQL Server database with FILESTREAM data.

FILESTREAM data restores to the database folder during redirected restore

FILESTREAM data restores to the folder with the database files, even if the FILESTREAM data was originally in a different folder than the database files, when the following conditions are true:

- ◆ You are restoring the database to a different location than the original location at the time of the backup.
- ◆ You specify the new location by using the **Alternate database location** box on the **Restore Command Line Options** dialog box.

To avoid this issue, specify the restore location for both the database files and the FILESTREAM data in the **Set Destination** dialog box. [“Restoring to a different instance” on page 113](#) provides instructions.

Restore fails when the folder with FILESTREAM data is open

Restore of a database with FILESTREAM data fails when the folder with FILESTREAM data is open in a program such as Windows Explorer.

To avoid this issue, close the program that is accessing the folder with the FILESTREAM before you perform the restore. However, if the FILESTREAM folder is open and the restore fails, then perform the restore again and select the WITH REPLACE option to force the restore and bring the database online.

FILESTREAM datafiles appear as Rows Data in Avamar Administrator

FILESTREAM datafiles appear with a description of **Rows Data** on the **Restore** tab of the **Backup, Restore and Manage** window in Avamar Administrator. The description in Avamar Administrator is different than the **Filestream data** description that appears in SQL Server Management Studio.

SQL Server Analysis Service (SSAS) requirements

The Avamar Plug-in for SQL Server cannot back up SSAS databases. However, you can back up the SSAS databases by using SQL Server Management Studio (SSMS) or the XMLA Query Editor, and then use the Avamar Client for Windows to back up the flat file that results from the SSMS or XMLA query backup.

The Microsoft website provides instructions on using SSMS or an XMLA query to

Considerations for the number of databases

The Avamar Management Console Server (MCS) may not be able to display all databases when you browse for data to back up or restore on a SQL Server installation with many databases. When this occurs, a Browse Timeout message appears and enables you to either set a new time limit in seconds or to view partial results. If you view partial results, no entries appear. The Avamar Plug-in for SQL Server does not support this option.

Checking the User Account Control setting on Microsoft Windows

The User Account Control (UAC) feature limits application software to standard user privileges. You must provide administrator privileges for certain tasks, such as installing software. UAC is enabled by default.

If you start an Avamar client or plug-in installer without administrator privileges on a computer with UAC enabled, then the software does not install correctly.

You can disable or bypass UAC. The installation procedures in this chapter provide one method to bypass UAC. Other methods and additional information are available in the Microsoft documentation.

Preparing a stand-alone server

1. Install and configure Microsoft SQL Server.
2. Install Microsoft .NET Framework 4. This software is required to install the Avamar Plug-in for SQL Server. Search the Microsoft Download Center for “Microsoft .NET Framework 4” to find downloads and additional information.
3. If you are using Microsoft SQL Server 2012, then add the Windows system service account to the SQL Server administrators group. [“Adding the Windows system service account as an administrator” on page 41](#) provides instructions.

Preparing a cluster

1. Install and prepare the cluster.
2. If you are installing SQL Server 2008 or SQL Server 2008 R2 in a Windows Server 2012 cluster, enable the COM-based `MsClust.dll` library on each node.

If you do not enable this library on the cluster node, then the SQL Server setup on the node fails with an error that cluster service verification failed.

To enable the library, enable the Failover Cluster Automation Server feature by using one of the following methods on each node in the cluster:

- In Server Manager, expand **Remote Server Administration Tools > Feature Administration Tools > Failover Clustering Tools**, and then select the **Failover Cluster Automation Server** checkbox.
- Run the following Windows PowerShell cmdlet from an elevated command prompt:

```
add-windowsfeature RSAT-Clustering-AutomationServer
```

3. If you are installing SQL Server 2008 or SQL Server 2008 R2 in a Windows Server 2012 cluster, install SQL Server 2008 SP3 or later, or SQL Server 2008 R2 SP1 or later.

Otherwise, a “Windows Server 2003 FILESTREAM Hotfix Check” error appears when you install and configure SQL Server in the cluster.

To ensure that you are installing the service pack version of SQL Server, you may need to perform one of the following steps:

- Download the correct service pack package, and run the service pack setup on each of the cluster nodes to pre-patch the nodes with setup binaries.

The setup process closes automatically after you install setup support files. You must reopen the setup wizard to perform the SQL Server installation.

NOTICE

If the SQL Support Files from SQL Server 2008 or SQL Server 2008 R2 are currently installed on the node, uninstall SQL Server before you run the service pack setup to install latest SQL Support Files.

- Create a slipstream package that includes the service pack, and use that to install the service pack version of SQL Server on each node.

Microsoft knowledgebase article 955392, “How to update or slipstream an installation of SQL Server 2008,” which is available on the Microsoft website, provides instructions on creating a slipstream package.

4. Install and configure Microsoft SQL Server on each node.
5. Install Microsoft .NET Framework 4 on each node. This software is required to install the Avamar Plug-in for SQL Server. Search the Microsoft Download Center for “Microsoft .NET Framework 4” to find downloads and additional information.
Microsoft .NET Framework 4 is pre-installed on Windows Server 2012.
6. If you are using SQL Server 2012, then add the Windows system service account to the SQL Server administrators group on each node. [“Adding the Windows system service account as an administrator” on page 41](#) provides instructions.

Downloading the software

1. Log in to the computer that is running SQL Server as an administrator.
2. Open a web browser and type the following URL:
`http://Avamar_server`
where *Avamar_server* is the DNS name or IP address of the Avamar server.
The **EMC Avamar Web Restore** page appears.
3. Click **Downloads**.
The **Downloads** list appears.
4. Click + next to the operating system headings until the applicable software installation packages appear.
5. Click the Avamar Client for Windows installation package:
 - `AvamarClient-windows-x86-version.msi` (32-bit)
 - `AvamarClient-windows-x86_64-version.msi` (64-bit)
 where *version* is the Avamar client version.
6. Save the Windows client installation package to a temporary folder.

7. Click the SQL Server plug-in installation package:
 - AvamarSQL-windows-x86-*version*.msi (32-bit)
 - AvamarSQL-windows-x86_64-*version*.msi (64-bit).
 where *version* is the Avamar client version.
8. Save the SQL Server plug-in installation package to a temporary folder.
9. Click the Avamar Config Checker installation package:
 - Avamar_ConfigChecker_win_x86.zip (32-bit)
 - Avamar_ConfigChecker_win_x64.zip (64-bit)
10. Save the Config Checker installation package to a temporary folder.

Verifying the environment

Use the Avamar Config Checker for Microsoft Windows to verify that you correctly configured the Windows environment for Avamar backup and recovery.

The Config Checker checks the configuration for problems that can lead to installation, backup, or recovery failures. These failures can affect the operating system for the application host or SQL Server.

The Config Checker supports only English language operating systems.

To use the Avamar Config Checker to verify the environment before you install Avamar:

1. Unzip the Avamar Config Checker installation package, and run the setup program to install the software.
2. Start the Config Checker:
 - On Windows Server 2012, open the **Start** screen and select **Avamar Config Checker**.
 - On Windows Server 2008 or Windows Server 2003, open the **Start** menu and select **Program Files > EMC Config Checker > Avamar Config Checker**.
3. Click **Next** on the welcome page.

The **Avamar Application and User Settings** page appears.
4. Select the version number from the **Avamar version** list.
5. In the application list, select the checkbox next to the applications on the client computer, and specify credentials, if required.
6. Click **Next**.
7. Review the summary information.
8. Click **Run Tests**.

When the verification completes, the **Config Check Results** window appears.
9. Save or open the results in HTML format.

NOTICE

Save the test results before exiting the wizard. Otherwise, the results are lost and you must rerun the Config Checker to view them.

10. Click **Finish** to exit the wizard.
11. Review the HTML result file, and correct all the checks listed under Failure.
12. Rerun the Config Checker to ensure that all the checks are successful.

You also can run the Avamar Config Checker after you install Avamar software on the client computer.

The *EMC Avamar Config Checker for Microsoft Windows Technical Note*, available on EMC Online Support at <https://support.EMC.com>, provides troubleshooting information and details about the results that appear for each application.

Installing the Avamar client software

You can install the Avamar client software on a stand-alone server or in a cluster environment.

Installation road maps

The steps to install the Avamar software for SQL Server depend on whether the computer that is running SQL Server is a standalone server or in a cluster environment.

Stand-alone installation road map

To install the Avamar client software on a stand-alone server:

1. Install the Avamar Client for Windows on the computer that is running SQL Server. [“Installing the Avamar Client for Windows” on page 50](#) provides instructions.
2. Install the Avamar Plug-in for SQL Server on the computer that is running SQL Server. [“Installing the Avamar Plug-in for SQL Server” on page 51](#) provides instructions.
3. Register the computer that is running SQL Server as a client with the Avamar server. [“Registering the client” on page 52](#) provides instructions.

Cluster installation road map

To install the Avamar software in a cluster environment:

1. Install the Avamar Client for Windows in the same folder on each node in the cluster. [“Installing the Avamar Client for Windows” on page 50](#) provides instructions.
2. Install the Avamar Plug-in for SQL Server on each node in the cluster. [“Installing the Avamar Plug-in for SQL Server” on page 51](#) provides instructions.
3. Register each node in the cluster with the Avamar server. [“Registering the client” on page 52](#) provides instructions.
4. Use the Cluster Configuration Tool to configure the Avamar cluster client:
 - Follow the steps in [“Configuring the cluster client in a failover cluster” on page 52](#) for SQL Server in a failover cluster.

- Follow the steps in [“Configuring the cluster client for an AlwaysOn availability group” on page 54](#) for a SQL Server AlwaysOn availability group.

Installing the Avamar Client for Windows

1. Log in to the computer that is running SQL Server as an administrator.
2. Go to the temporary folder that contains the Avamar installation files that you downloaded in [“Downloading the software” on page 47](#).
3. Start the installer:
 - If UAC is disabled, double-click the installation file to open it.
 - If UAC is enabled, perform the following steps:
 - a. In Windows, right-click the **Command Prompt** icon and select **Run as administrator**.
 - b. In the **Command Prompt** window, change the working directory to the location of the installation package by typing:

```
cd install_path
```

where *install_path* is the full path of the temporary folder that contains the installation package.

- c. Type one of the following commands to start the installer:

```
msiexec /i AvamarClient-windows-x86-version.msi
```

```
msiexec /i AvamarClient-windows-x86_64-version.msi
```

where *version* is the Avamar client version.

[“Checking the User Account Control setting on Microsoft Windows” on page 46](#) provides details on UAC.

The installation wizard opens at the welcome page.

4. Click **Next**.
The **End-User License Agreement** page appears.
5. Review the license agreement.
6. Select the checkbox to accept the terms in the license agreement, and click **Next**.
The **Custom Setup** page appears.
7. Ensure that **Avamar Client User Interface** is selected for installation.
8. (Optional) To specify a folder for the Avamar client installation, click **Browse** and select a location.
9. Click **Next** on the **Custom Setup** page.
The **Ready to Install EMC Avamar for Windows** page appears.
10. Click **Install**.
11. When the installation completes, click **Finish**.

Installing the Avamar Plug-in for SQL Server

1. Log in to the computer that is running SQL Server as an administrator.
2. Go to the temporary folder that contains the Avamar installation files that you downloaded in [“Downloading the software” on page 47](#).
3. Start the SQL Server plug-in installation:
 - If UAC is disabled, double-click the installation file to open it.
 - If UAC is enabled, perform the following steps:
 - a. In Windows, right-click the **Command Prompt** icon and select **Run as administrator**.
 - b. In the **Command Prompt** window, change the working directory to the location of the installation package by typing:

```
cd install_path
```

where *install_path* is the full path of the temporary folder that contains the installation package.

- c. Type one of the following commands to start the installer:

```
msiexec /i AvamarSQL-windows-x86-version.msi
```

```
msiexec /i AvamarSQL-windows-x86_64-version.msi
```

where *version* is the Avamar plug-in version.

The installation wizard opens at the welcome page.

4. Click **Next**.

The **Ready to Install EMC Avamar Backup Plug-in for SQL Server** page appears.

Note: The SQL Server plug-in automatically installs to the same folder as the Avamar Client for Windows.


5. Click **Install**.
6. When the installation completes, click **Finish**.

Registering the client

Before you can back up or restore SQL Server data, you must register the computer that is running SQL Server as a client with the Avamar server.

In a cluster environment, register all nodes in the cluster as Avamar clients.

To register the client:

1. Log in to the computer that is running SQL Server.
2. Right-click the Avamar client system tray icon .

A menu appears.
3. Select **Manage > Activate Client**.

The **Activate Client Setup** dialog box appears.
4. In the **Administrator Server Address** box, type the network hostname defined in DNS for the Avamar server.
5. In the **Administrator Server Port** box, specify the port on the Avamar server for client/server communication. The default value is 28001.
6. In the **Client Domain** box, type the name of the Avamar domain for the client.

Do not use a slash (/) as the first character when you type a subdomain. If you use a slash, an error occurs and you cannot register the client.
7. Click **Activate**.

A confirmation message appears.
8. Click **OK**.

Configuring the cluster client in a failover cluster

1. Log in to the active node in the cluster as a domain administrator. The account must also be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool:
 - On Windows Server 2012, open the **Start** screen and select **Cluster Configuration Tool**.
 - On Windows Server 2008 or Windows Server 2003, open the **Start** menu and select **Program Files > EMC Avamar > Cluster Configuration Tool**.

The welcome page appears.
3. Click **Next**.

The **Plug-Ins** page appears.
4. Select **SQL** and click **Next**.

The **Cluster Nodes** page appears with a list of nodes and their status.

5. Ensure that the environment meets the following requirements:
 - The status for each SQL Server node is Up.
 - The installation status of the Windows client software for each node is Installed.
 - The installation status of the SQL Server plug-in on each node is Installed.
6. Click **Next**.

The **Operations** page appears.
7. Select **Configure a new cluster client for all nodes** and click **Next**.

The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.
8. Ensure that the environment meets all prerequisites on the **Prerequisites** page.

If the environment does not meet a prerequisite, then exit the wizard, resolve the issue, and restart the wizard.
9. Select the IP version that the environment uses, and then click **Next**.

The **SQL Settings** page appears.
10. Select the cluster group, service, or role for the cluster client from the list:
 - On Windows Server 2012, select the cluster role for the cluster client from the **Cluster role for cluster client** list.
 - On Windows Server 2008, select the cluster service for the cluster client from the **Cluster service for cluster client** list.
 - On Windows Server 2003, select the cluster group for the cluster client from the **Cluster group for cluster client** list.

The name of the cluster client appears in the **Cluster client name** box.
11. Select the shared volume for the cluster client from the **Shared volume for the cluster client** list.
12. Specify whether to automatically bring the cluster client online and activate it with the Avamar server by selecting or clearing the **Bring the cluster client online and activate it with the Avamar server** checkbox.
13. Click **Next**.

The **Server Settings** page appears.
14. Specify the settings for the Avamar server:
 - a. Type either the DNS name of the Avamar server in the **Name** box or the IP address in the **IPv4/IPv6 address** box.
 - b. Type the name of the Avamar domain for the cluster client in the **Avamar client domain for the cluster client** box.
 - c. Type the data port for Avamar client/server communication in the **Port number** box.

- d. Type the path to the `var` folder for the cluster client in the **Cluster client's var directory** box, or click **Browse** to select a location.

The `var` folder stores the cluster client configuration and log files. All nodes in the cluster must have write access to this location.

NOTICE

Select a volume that the cluster owns instead of a remote pathname on the network.

15. Click **Next**.

The **Summary** page appears.

16. Review the settings that you specified in the wizard, and click **Configure**.

The **Progress** page provides the status of the configuration. When the configuration is complete, the **Results** page appears.

17. Click **Close**.

Configuring the cluster client for an AlwaysOn availability group

1. Log in to the cluster node with the primary replica as a domain administrator. The account must also be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool:
 - On Windows Server 2012, open the **Start** screen and select **Cluster Configuration Tool**.
 - On Windows Server 2008, open the **Start** menu and select **Program Files > EMC Avamar > Cluster Configuration Tool**.

The welcome page appears.

3. Click **Next**.

The **Plug-Ins** page appears.

4. Select **SQL AlwaysOn** and click **Next**.

The **Cluster Nodes** page appears with a list of nodes and their status.

5. Ensure that the environment meets the following requirements:

- The status for each SQL Server node is Up.
- The installation status of the Windows client software for each node is Installed.
- The installation status of the SQL Server plug-in on each node is Installed.

6. Click **Next**.

The **Operations** page appears.

7. Select **Configure a new cluster client for all nodes** and click **Next**.

The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.

8. Ensure that the environment meets all prerequisites on the **Prerequisites** page.

If the environment does not meet a prerequisite, then exit the wizard, resolve the issue, and restart the wizard.

9. Select the IP version that the environment uses, and then click **Next**.

The **SQL AlwaysOn Settings** page appears.

10. Select the cluster group, service, or role for the cluster client from the list:

- On Windows Server 2012, select the cluster role for the cluster client from the **Cluster role for cluster client** list.
- On Windows Server 2008, select the cluster service for the cluster client from the **Cluster service for cluster client** list.

The name of the cluster client appears in the **Cluster client name** box.

Note: You must configure an availability group listener for each availability group. Do not configure a cluster client for an availability group that does not have a listener.

11. Specify whether to automatically bring the cluster client online and activate it with the Avamar server by selecting or clearing the **Bring the cluster client online and activate it with the Avamar server** checkbox.

12. Click **Next**.

The **Server Settings** page appears.

13. Specify the settings for the Avamar server:

- a. Type either the DNS name of the Avamar server in the **Name** box or the IP address in the **IPv4/IPv6 address** box.
- b. Type the name of the Avamar domain for the cluster client in the **Avamar client domain for the cluster client** box.
- c. Type the data port for Avamar client/server communication in the **Port number** box.
- d. Type the path to the `var` folder for the cluster client in the **Cluster client's var directory** box, or click **Browse** to select a location.

The `var` folder stores the cluster client configuration and log files. All nodes in the cluster must have write access to this location.

NOTICE

Select a volume that the cluster owns instead of a remote pathname on the network.

14. Click **Next**.

The **Summary** page appears.

15. Review the settings that you specified in the wizard, and click **Configure**.

The **Progress** page provides the status of the configuration. When the configuration is complete, the **Results** page appears.

16. Click **Close**.

Upgrading the Avamar client software

The steps to upgrade Avamar on a SQL Server client depend on whether the installation is on a stand-alone server or in a cluster.

NOTICE

If you upgrade the Avamar Plug-in for SQL Server to release 6.1 or 7.0, then you must also upgrade the Avamar server software to release 6.1 or 7.0. Do not use a SQL Server plug-in running release 6.0 or 5.0 to back up to or restore from an Avamar server running release 6.1 or 7.0. Also, do not use a SQL Server plug-in running release 6.1 or 7.0 to back up to or restore from an Avamar server running release 6.0 or 5.0.

Upgrading on a stand-alone server

When you upgrade Avamar client and plug-in software on a stand-alone server, you do not need to uninstall earlier versions before you install a new version. The installation determines that an earlier version is installed, and then prompts you to upgrade to the new version or remove the current version.

To upgrade Avamar client and plug-in software on a stand-alone server:

1. Ensure that the environment meets all system requirements for the new version. [“Preparing to install the SQL Server plug-in” on page 40](#) provides information.
2. Upgrade the Avamar Client for Windows by running the Windows client installation wizard for the new version on the Windows server. [“Installing the Avamar Client for Windows” on page 50](#) provides instructions.
3. Upgrade the Avamar Plug-in for SQL Server by running the plug-in installation wizard for the new version on the Windows server. [“Installing the Avamar Plug-in for SQL Server” on page 51](#) provides instructions.
4. Perform a full backup of all SQL Server data to ensure that you have a current full backup with the new version.

NOTICE

If you store backups on a Data Domain system, then you must perform a full backup after you upgrade the Avamar Plug-in for SQL Server to a newer version. Otherwise, restore fails when you restore from either a tail-log backup, an incremental backup, or a differential backup that occurs after the upgrade.

Upgrading in a cluster

When you upgrade Avamar client and plug-in software in a cluster, you must uninstall the current Avamar clients and plug-ins from each node, and then install the new version.

To upgrade Avamar client and plug-in software in a cluster:

1. Ensure that the environment meets all system requirements for the new version. [“Preparing to install the SQL Server plug-in” on page 40](#) provides information.

2. Uninstall the current version of the Avamar client and plug-in:
 - a. Use the Cluster Configuration Tool to uninstall the Avamar cluster client.
 - b. Uninstall the earlier version of the Avamar Plug-in for SQL Server.
 - c. Uninstall the earlier version of the Avamar Client for Windows.

The plug-in guide for the earlier version provides instructions for each of these steps.

3. Install the new version of the Avamar client and plug-in:
 - a. Install the Avamar Client for Windows in the same folder on each node in the cluster. [“Installing the Avamar Client for Windows” on page 50](#) provides instructions.
 - b. Install the Avamar Plug-in for SQL Server in the same folder on each node in the cluster. [“Installing the Avamar Plug-in for SQL Server” on page 51](#) provides instructions.
 - c. Register each node in the cluster with the Avamar server. [“Registering the client” on page 52](#) provides instructions.
 - d. Configure the Avamar Cluster Client:
 - Follow the steps in [“Configuring the cluster client in a failover cluster” on page 52](#) for SQL Server in a failover cluster.
 - Follow the steps in [“Configuring the cluster client for an AlwaysOn availability group” on page 54](#) for a SQL Server AlwaysOn availability group.
4. Perform a full backup of all SQL Server data to ensure that you have a current full backup with the new version.

NOTICE

If you store backups on a Data Domain system, then you must perform a full backup after you upgrade the Avamar Plug-in for SQL Server to a newer version. Otherwise, restore fails when you restore from either a tail-log backup, an incremental backup, or a differential backup that occurs after the upgrade.

Uninstalling the Avamar client software

You can uninstall Avamar client and plug-in software on a stand-alone server or in a cluster.

Uninstall road map

To uninstall Avamar client and plug-in software from a SQL Server client on either a stand-alone server or in a cluster:

1. (Cluster only) Uninstall the Avamar cluster client:
 - Follow the steps in [“Uninstalling the cluster client in a failover cluster”](#) on page 58 for SQL Server in a failover cluster.
 - Follow the steps in [“Uninstalling the cluster client for an AlwaysOn availability group”](#) on page 59 for a SQL Server AlwaysOn availability group.
2. Uninstall the Avamar Plug-in for SQL Server. [“Uninstalling the Avamar Plug-in for SQL Server”](#) on page 60 provides instructions.
3. Uninstall the Avamar Client for Windows. [“Uninstalling the Avamar Client for Windows”](#) on page 60 provides instructions.
4. (Cluster only) Repeat [step 2](#) and [step 3](#) on each node.

Uninstalling the cluster client in a failover cluster

1. Log in to the active node in the cluster as a domain administrator. The account must also be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool:
 - On Windows Server 2012, open the **Start** screen and select **Cluster Configuration Tool**.
 - On Windows Server 2008 or Windows Server 2003, open the **Start** menu and select **Program Files > EMC Avamar > Cluster Configuration Tool**.

The welcome page appears.

3. Click **Next**.
The **Plug-Ins** page appears.
4. Select **SQL** and click **Next**.
The **Cluster Nodes** page appears with a list of nodes and their status.
5. Ensure that the status of each SQL Server node is Up, and then click **Next**.
The **Operations** page appears.
6. Select **Remove the cluster client from all nodes**, and then click **Next**.
The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.
7. Ensure that the environment meets all prerequisites on the page, and click **Next**.
The **Uninstall SQL** page appears.

8. Select the cluster role, service, or group that contains the cluster client from the **Cluster role/service/group for cluster client** list, and then click **Next**.
9. Select the cluster client from the **Cluster client name** list.
10. Select the shared volume for the cluster client from the **Shared volume for the cluster client** list.
11. Click **Next**.
The **Summary** page appears.
12. Review the settings that you specified in the wizard, and click **Uninstall**.
The **Progress** page provides the status of the uninstall. When the uninstall is complete, the **Results** page appears.
13. Click **Close**.

Uninstalling the cluster client for an AlwaysOn availability group

1. Log in to the cluster node with the primary replica as a domain administrator. The account must also be a member of the local Administrators group on each cluster node.
2. Start the Cluster Configuration Tool:
 - On Windows Server 2012, open the **Start** screen and select **Cluster Configuration Tool**.
 - On Windows Server 2008, open the **Start** menu and select **Program Files > EMC Avamar > Cluster Configuration Tool**.
 The welcome page appears.
3. Click **Next**.
The **Plug-Ins** page appears.
4. Select **SQL AlwaysOn** and click **Next**.
The **Cluster Nodes** page appears with a list of nodes and their status.
5. Ensure that the status of each SQL Server node is Up, and click **Next**.
The **Operations** page appears.
6. Select **Remove the cluster client from all nodes**, and then click **Next**.
The **Prerequisites** page appears. A check mark next to a prerequisite indicates that the environment meets the prerequisite.
7. Ensure that the environment meets all prerequisites on the page, and click **Next**.
The **Uninstall Settings** page appears.
8. Select the cluster role or service that contains the cluster client from the **Cluster role/service for cluster client** list, and then click **Next**.
The **Summary** page appears.
9. Review the settings that you specified in the wizard, and click **Uninstall**.

The **Progress** page provides the status of the uninstall. When the uninstall is complete, the **Results** page appears.

10. Click **Close**.

Uninstalling the Avamar Plug-in for SQL Server

To uninstall the Avamar Plug-in for SQL Server:

- ◆ On Windows Server 2012 or Windows Server 2008, use **Programs and Features**.
- ◆ On Windows Server 2003, use **Add/Remove Programs**.

Uninstalling the Avamar Client for Windows

To uninstall the Avamar Client for Windows:

- ◆ On Windows Server 2012 or Windows Server 2008, use **Programs and Features**.
- ◆ On Windows Server 2003, use **Add/Remove Programs**.

CHAPTER 3

Backup

The following topics describe how to use the Avamar Plug-in for SQL Server to perform on-demand and scheduled backups of SQL Server databases:

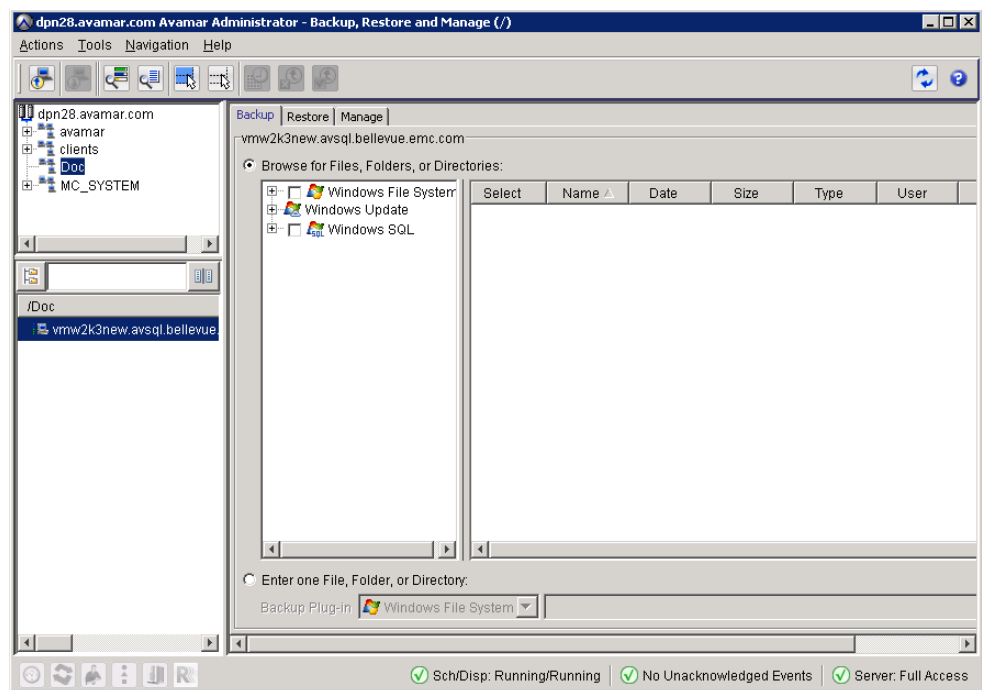
- ◆ [Performing on-demand backups.....](#) 62
- ◆ [Performing scheduled backups](#) 67
- ◆ [Setting backup options](#) 79
- ◆ [Monitoring backups](#) 93
- ◆ [Troubleshooting backups](#) 94
- ◆ [Canceling backups.....](#) 94

Performing on-demand backups

To perform an on-demand backup of one or more SQL Server instances or databases:

1. In Avamar Administrator, click the **Backup & Restore** launcher button.
The **Backup, Restore and Manage** window appears.
2. In the domain tree, select the domain for the client.
3. From the list of clients, select the computer that is running SQL Server. Keep the following points in mind when you select a client:
 - If you are backing up databases in an AlwaysOn availability group, then select the cluster client for the availability group listener.
 - If SQL Server is installed in a failover cluster and you are not backing up a database in an availability group, then select the cluster client for the virtual server.
 - You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
4. Click the **Backup** tab.

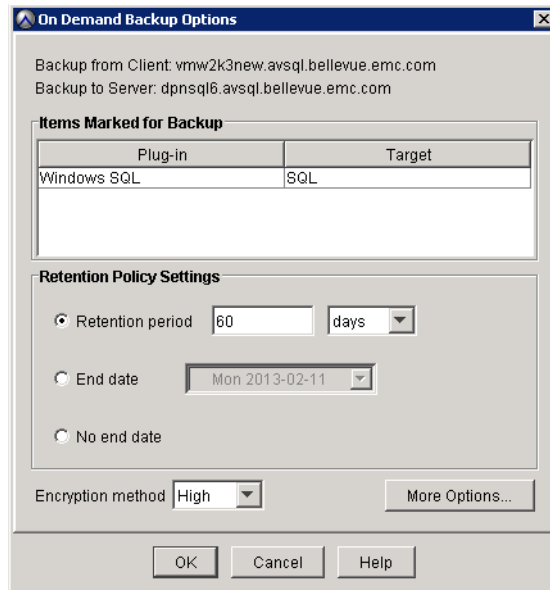
A list of plug-ins installed on the selected client appears in the left pane of the **Backup** tab.



5. Select the data to back up using the steps in one of the following topics:
 - [“Selecting all data on a client for backup” on page 65](#)
 - [“Browsing to instances or databases for backup” on page 65](#)
 - [“Typing the name of an instance or database for backup” on page 66](#)

6. Select **Actions > Back Up Now**.

The **On Demand Backup Options** dialog box appears.



7. Select the backup retention setting:

- To automatically delete this backup from the Avamar server after a specific amount of time, select **Retention period** and then specify the number of days, weeks, months, or years for the retention period.
- To automatically delete this backup from the Avamar server on a specific calendar date, select **End date** and browse to that date on the calendar.
- To keep this backup for as long as this client remains active in the Avamar server, select **No end date**.

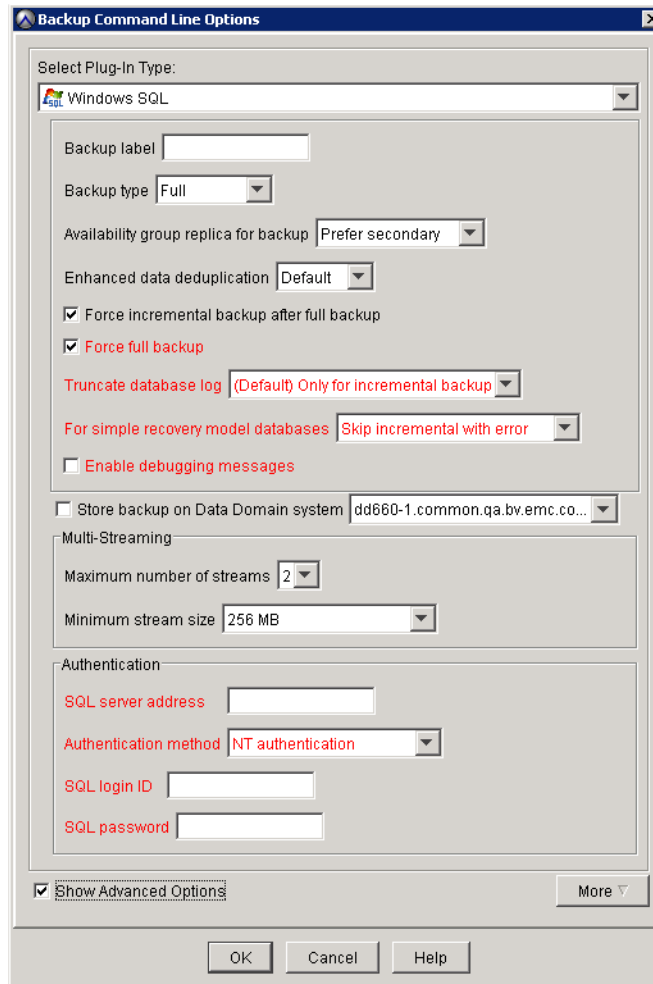
8. Select the encryption method to use for client/server data transfer during this backup.

The encryption technology and bit strength for a client/server connection depends on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

9. Click **More Options**.

The **Backup Command Line Options** dialog box appears.

10. Select the **Show Advanced Options** checkbox to view advanced options, which appear in red.



11. Set the plug-in options. [“Setting backup options”](#) on page 79 provides details on each of the options.
12. Click **OK** on the **Backup Command Line Options** dialog box.
13. Click **OK** on the **On Demand Backup Options** dialog box.

The **On Demand Backup Request** dialog box indicates that the backup started.

14. Click **Close**.

Selecting SQL Server data for an on-demand backup

You can back up all SQL Server data on a client, or you can select one or more instances or databases to back up.

Selecting all data on a client for backup

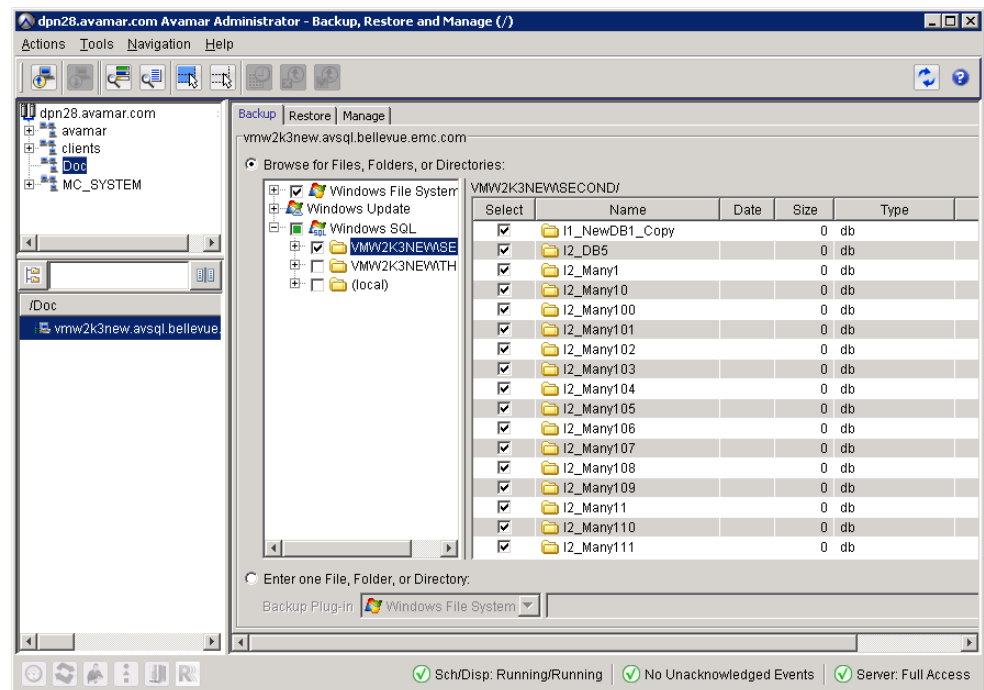
1. On the **Backup** tab of the **Backup, Restore and Manage** window, select **Browse for Files, Folders, or Directories**.
2. Select the checkbox next to the **Windows SQL** plug-in node.

Browsing to instances or databases for backup

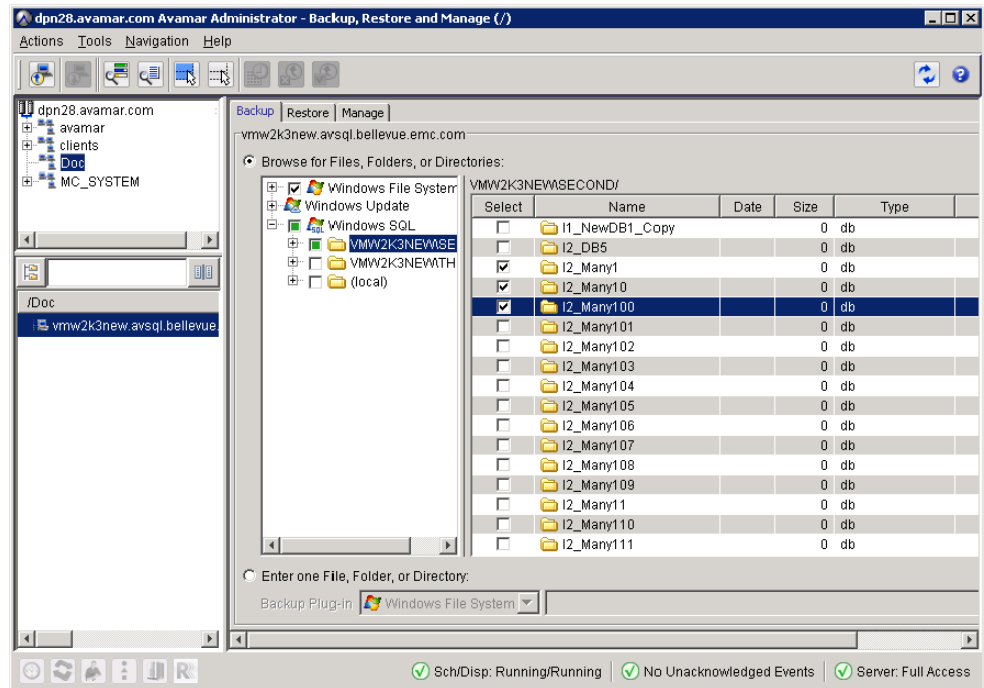
1. On the **Backup** tab of the **Backup, Restore and Manage** window, select **Browse for Files, Folders, or Directories**.
2. Expand the **Windows SQL** plug-in node.
3. Under the **Windows SQL** plug-in node, select the SQL Server instance that contains the databases to back up.

A list of the databases for that instance appears in the right pane of the **Select for Backup** tab.

4. Select the data to back up:
 - To back up all databases in the instance, select the checkbox next to the instance.



- To back up individual databases, select the checkbox next to the databases in the right pane of the **Select for Backup** tab.



Typing the name of an instance or database for backup

- On the **Backup** tab of the **Backup, Restore and Manage** window, select **Enter one File, Folder, or Directory**.
- From the **Backup Plug-in** list, select **Windows SQL**.
- In the box next to the **Backup Plug-in** list, type the instance or database:
 - To back up the local instance, type `(local)`.
 - To back up a database in the local instance, type `(local)/database/`.
 - To back up a named instance, type `client\instance/`.
 - To back up a database in a named instance, type `client\instance/database/`.
 - To back up a database if there is only one instance on the client and it is not the local instance, type `client/database/`.

where *client* is the name of the computer that is running SQL Server, *instance* is the name of the named instance, and *database* is the name of the database.

Performing scheduled backups

To perform scheduled backups of one or more SQL Server instances or databases:

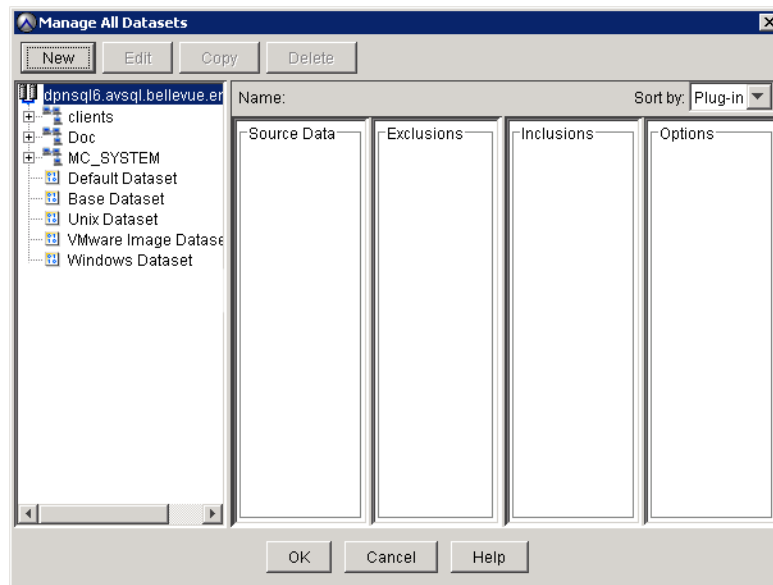
1. Create a dataset for the backups. [“Creating a dataset” on page 67](#) provides instructions.
2. Create a group for the backups. [“Creating a group” on page 74](#) provides instructions. During the group creation process, you:
 - a. Assign the new dataset to the new group.
 - b. Assign a schedule to the new group.
 - c. Assign a retention policy to the new group.
 - d. Add the SQL Server client to the new group.
3. Enable scheduling for the group. [“Enabling scheduled backups” on page 78](#) provides instructions.

The *EMC Avamar Administration Guide* provides more information about groups, group policy, datasets, schedules, and retention policies.

Creating a dataset

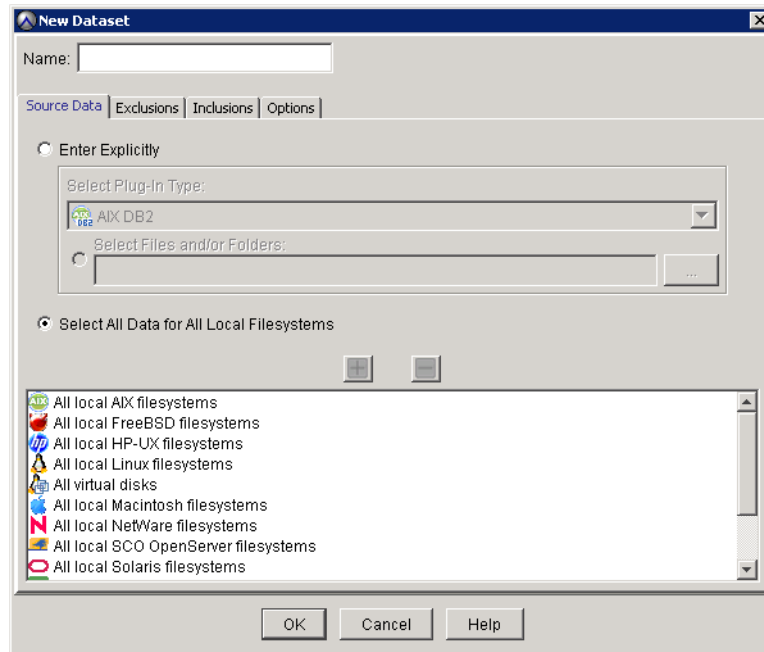
1. In Avamar Administrator, select **Tools > Manage Datasets**.

The **Manage All Datasets** window appears.



2. Click **New**.

The **New Dataset** dialog box appears.

3. In the **Name** box, type a name for the dataset.

The name can include alphanumeric characters (A-Z, a-z, 0-9) and the following special characters: period (.), hyphen (-), and underscore (_). Do not use Unicode characters or the following special characters: ` ~ ! @ # \$ % ^ & * () = + [] { } | \ / ; : ' " < > , ?

4. On the **Source Data** tab, select the data to include in the dataset. Follow the steps in one of the following topics:
 - [“Selecting data from all plug-ins on a client for scheduled backups” on page 70](#)
 - [“Selecting all SQL Server data on a client for scheduled backups” on page 70](#)
 - [“Selecting SQL Server instances or databases for scheduled backups” on page 70](#)
5. (Optional) Click the **Exclusions** tab and define databases or instances to exclude from the dataset. [“Specifying data to exclude from a dataset” on page 71](#) provides instructions.
6. (Optional) Click the **Inclusions** tab and define databases or instances to include in the dataset that otherwise would be excluded based on the selections on the **Exclusions** tab. [“Specifying excluded data to include” on page 73](#) provides instructions.

7. Click the **Options** tab and set plug-in options:
 - a. Select the **Windows SQL** plug-in from the **Select Plug-In Type** list.
The SQL Server plug-in options appear on the **Options** tab.
 - b. Select the **Show Advanced Options** checkbox to view advanced options, which appear in red.

The screenshot shows the 'New Dataset' dialog box with the 'Options' tab selected. The 'Select Plug-In Type' dropdown is set to 'Windows SQL'. The 'Backup label' field is empty. The 'Backup type' is set to 'Full'. The 'Availability group replica for backup' is set to 'Prefer secondary'. The 'Enhanced data deduplication' is set to 'Default'. The 'Force incremental backup after full backup' checkbox is checked. The 'Force full backup' checkbox is checked and highlighted in red. The 'Truncate database log' dropdown is set to '(Default) Only for incremental backup'. The 'For simple recovery model databases' dropdown is set to 'Skip incremental with error'. The 'Enable debugging messages' checkbox is unchecked. The 'Store backup on Data Domain system' checkbox is unchecked, with the path 'dd660-1.common.qa.bv.emc.co...' shown. The 'Multi-Streaming' section is expanded, showing 'Maximum number of streams' set to 2 and 'Minimum stream size' set to 256 MB. The 'Authentication' section is expanded, showing 'SQL server address', 'Authentication method' set to 'NT authentication', 'SQL login ID', and 'SQL password' fields. The 'Show Advanced Options' checkbox is checked, and the advanced options are displayed in red text. The 'Reset' and 'More' buttons are visible at the bottom right of the dialog box.

- c. Set the plug-in options. [“Setting backup options” on page 79](#) provides details on each option.
8. Click **OK**.

Selecting the data for a dataset

When you create a dataset for scheduled backups, you select the data to include in the backups:

- ◆ Data from all plug-ins installed on the client
- ◆ All SQL Server data on the client
- ◆ Specific SQL Server instances or databases

Selecting data from all plug-ins on a client for scheduled backups

To include data from all plug-ins installed on the client, select **Select All Data for All Local File Systems** on the **Source Data** tab of the **New Dataset** dialog box.

Selecting all SQL Server data on a client for scheduled backups

1. On the **Source Data** tab of the **New Dataset** dialog box, select **Enter Explicitly**.
2. Select **Windows SQL** from the **Select Plug-In Type** list.
3. Select the **SQL** option.

Selecting SQL Server instances or databases for scheduled backups

1. On the **Source Data** tab of the **New Dataset** dialog box, select **Enter Explicitly**.
2. Select **Windows SQL** from the **Select Plug-In Type** list.
3. Select the **Select Files and/or Folders** option.
4. Click ... (**Browse for files and/or folders**).

The **Select Files And/Or Folders** dialog box appears.

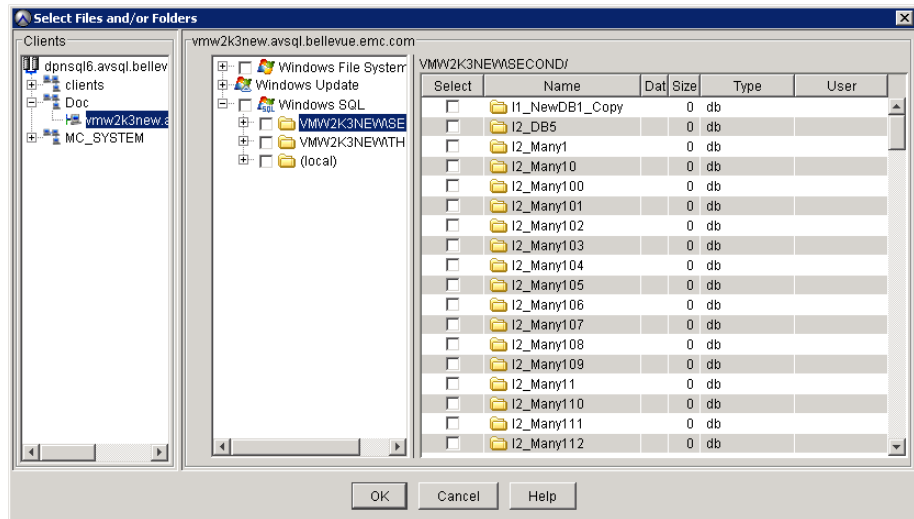
5. Select the client from the **Clients** tree in the left pane.

If the dataset is for backups of databases in an AlwaysOn availability group, then select the cluster client for the availability group listener.

If the dataset is for backups of databases on shared storage in a failover cluster, then select the cluster client for the virtual server.

6. In the middle pane of the **Select Files And/Or Folders** dialog box, expand the **Windows SQL** plug-in node.
7. Under the **Windows SQL** plug-in node, select the SQL Server instance that contains the databases to include.

A list of the databases for that instance appears in the right pane of the **Select Files And/Or Folders** dialog box.



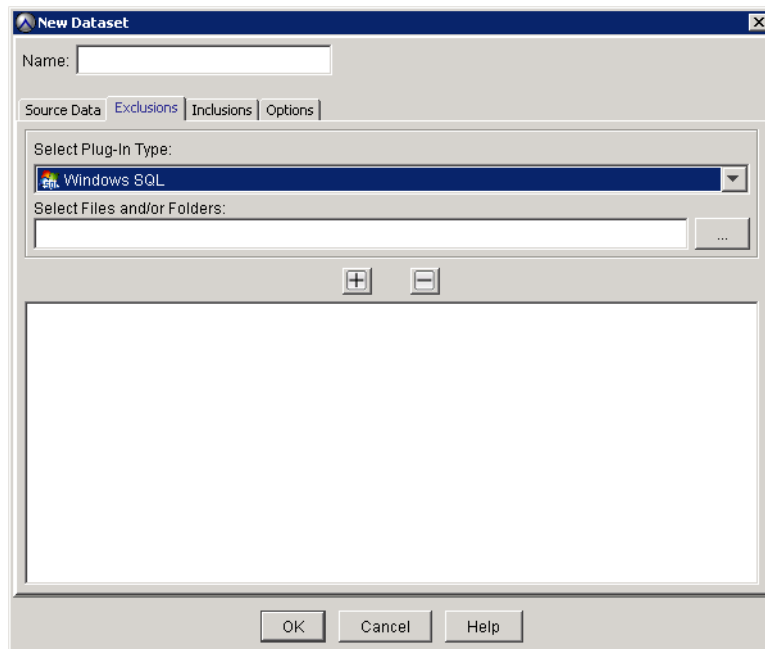
8. Select the instances or databases:

- To include all databases in an instance, select the checkbox next to the instance in the middle pane.
- To include individual databases, select the checkbox next to the databases in the right pane.

9. Click **OK**.

Specifying data to exclude from a dataset

1. From the **Select Plug-In Type** list on the **Exclusions** tab of the **New Dataset** dialog box, select **Windows SQL**.

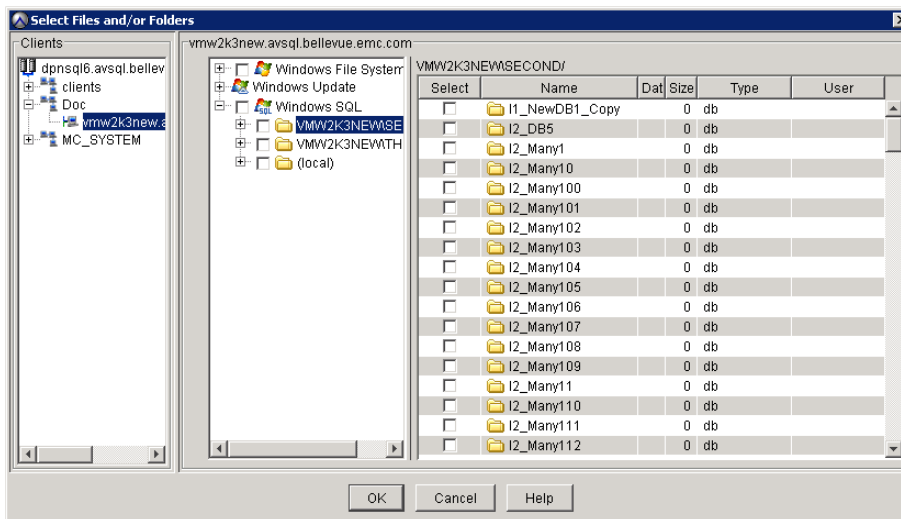


2. Click ... to browse to the instances or databases to exclude.

The **Select Files And/Or Folders** dialog box appears.

3. Select the client from the **Clients** tree in the left pane.
4. In the middle pane of the **Select Files And/Or Folders** dialog box, expand the **Windows SQL** plug-in node.
5. Under the **Windows SQL** plug-in node, select the SQL Server instance that contains the databases to exclude.

A list of the databases for that instance appears in the right pane of the **Select Files And/Or Folders** dialog box.

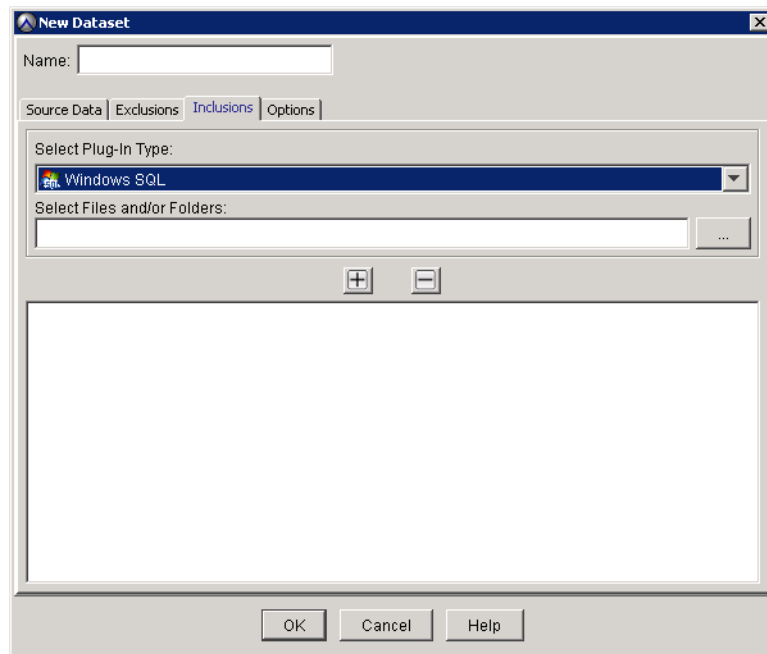


6. Select the data to exclude from the dataset:
 - To exclude all databases in the instance, select the checkbox next to the instance in the middle pane.
 - To exclude individual databases, select the checkbox next to the databases in the right pane.
7. Click **OK**.

The selected instances or databases appear on the **Exclusions** tab.

Specifying excluded data to include

1. From the **Select Plug-In Type** list on the **Inclusions** tab of the **New Dataset** dialog box, select **Windows SQL**.



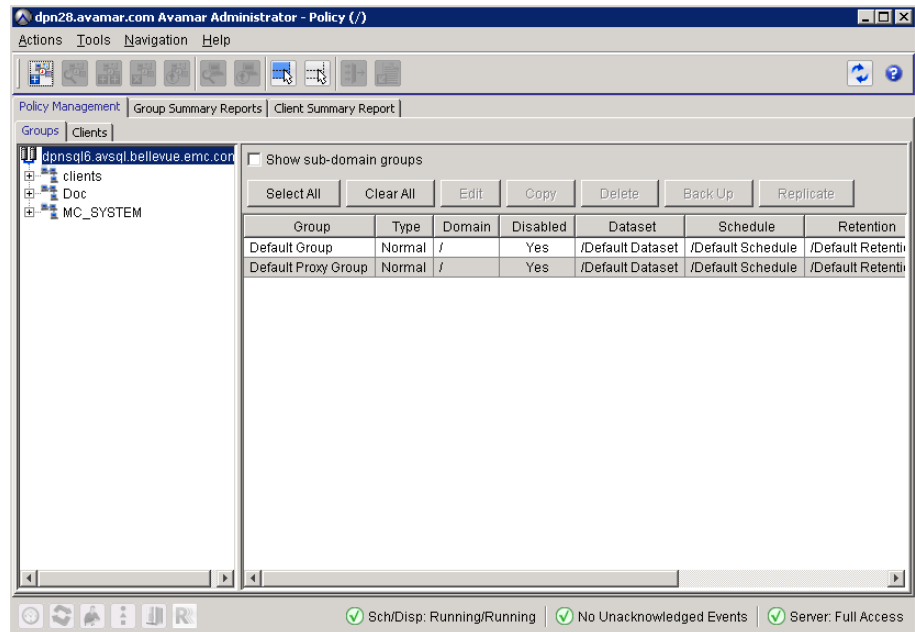
2. Click ... to browse to the instances or databases to include.
The **Select Files And/Or Folders** dialog box appears.
3. Select the client from the **Clients** tree in the left pane.
4. In the middle pane of the **Select Files And/Or Folders** dialog box, expand the **Windows SQL** plug-in node.
5. Under the **Windows SQL** plug-in node, select the SQL Server instance that contains the databases to include.
A list of the databases for that instance appears in the right pane of the **Select Files And/Or Folders** dialog box.
6. Select the data to include in the dataset that otherwise would be excluded based on the selections on the **Exclusions** tab:
 - To include all databases in the instance, select the checkbox next to the instance in the middle pane.
 - To include individual databases, select the checkbox next to the databases in the right pane.
7. Click **OK**.
The selected instances or databases appear on the **Inclusions** tab.

Creating a group

To create a group for scheduled SQL Server backups:

1. In Avamar Administrator, click the **Policy** tab.

The **Policy** window appears.



2. On the **Policy Management** tab, select the **Groups** tab.
3. In the left pane, select the Avamar domain for the group.
4. Select **Actions > New Group**.

The **New Group** wizard appears.

5. In the **Name** box, type a name for the new group.

The name can include alphanumeric characters (A-Z, a-z, 0-9) and the following special characters: period (.), hyphen (-), and underscore (_). Do not use Unicode characters or the following special characters: ` ~ ! @ # \$ % ^ & * () = + [] { } | \ / ; : ' " < > , ?

6. Clear the **Disabled** checkbox to use this group to perform scheduled client backups.

Selecting the checkbox disables backups for the group.

7. From the **Encryption method** list, select the encryption method for client/server network communications. All clients in the group use this encryption method unless you override the method at the client level.

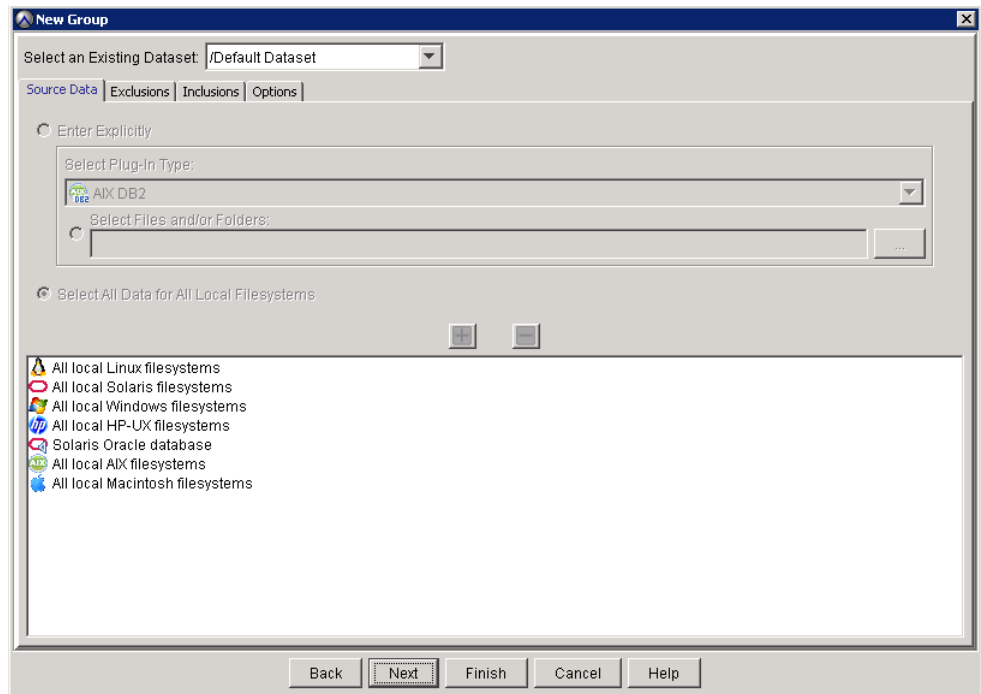
The encryption technology and bit strength for a client/server connection depends on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

8. Select whether to use the assigned schedule for the group or override the assigned schedule:

- To use the assigned schedule, leave the **Override Schedule** checkbox clear.
- To override the schedule, select **Override Schedule**, and then specify whether to skip the next scheduled backup entirely or to perform the next scheduled backup one time only by selecting **Skip Next Backup** or **Run Next Backup Once**.

9. Click **Next**.

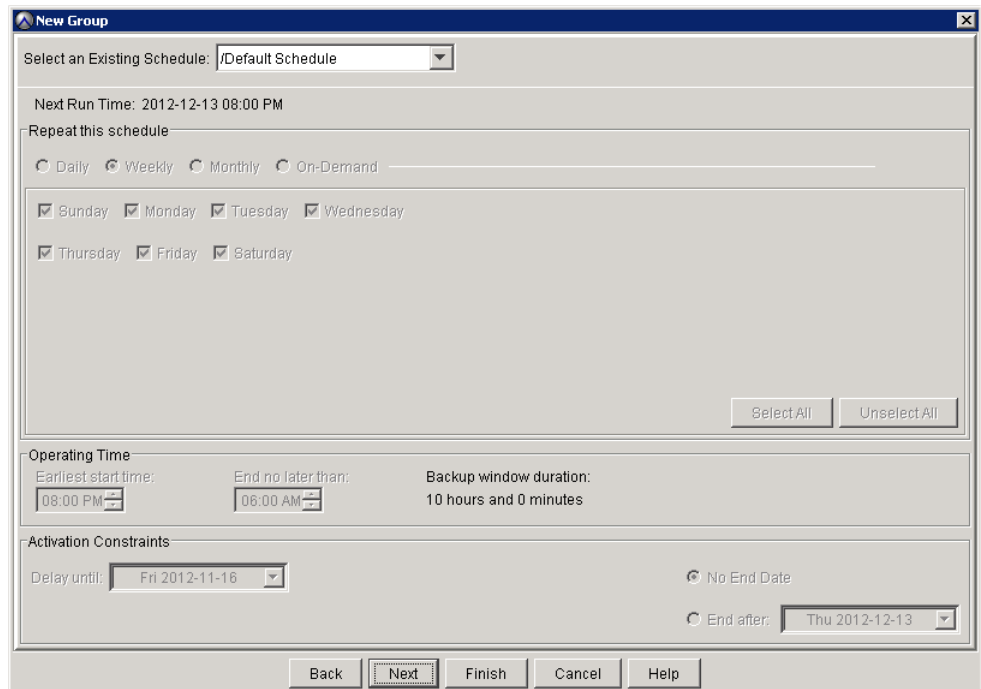
The next **New Group** wizard page appears with dataset information.



10. From the **Select An Existing Dataset** list, select the dataset that you defined during “Creating a dataset” on page 67.

11. Click **Next**.

The next **New Group** wizard page appears with schedule information.

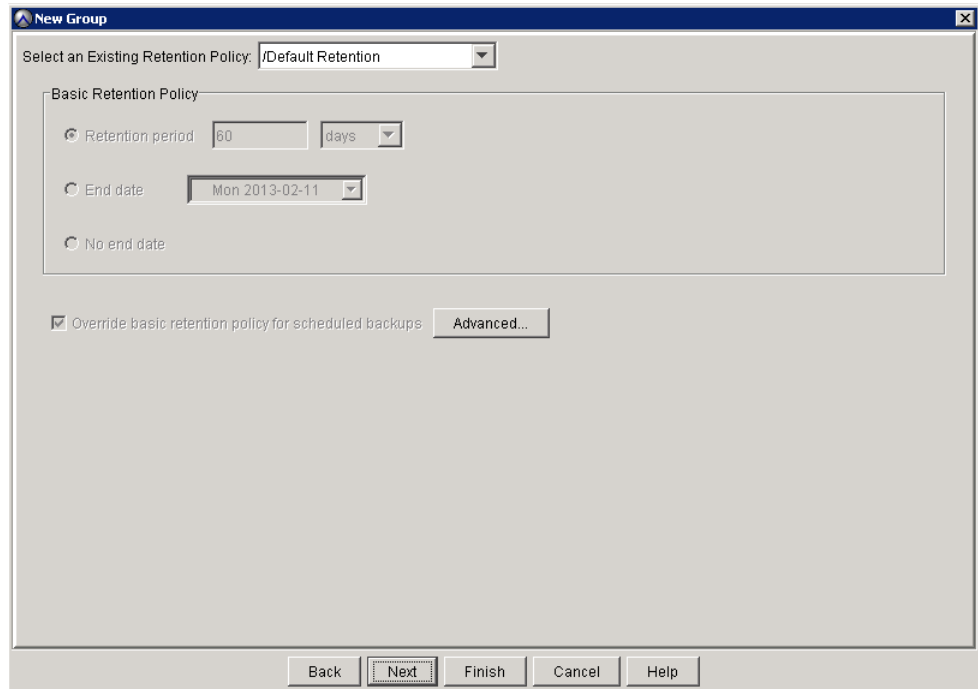


12. From the **Select An Existing Schedule** list, select a schedule for the group.

You cannot edit the schedules on this page. The *EMC Avamar Administration Guide* provides information about editing schedule properties.

13. Click **Next**.

The next **New Group** wizard page appears with retention policy information.



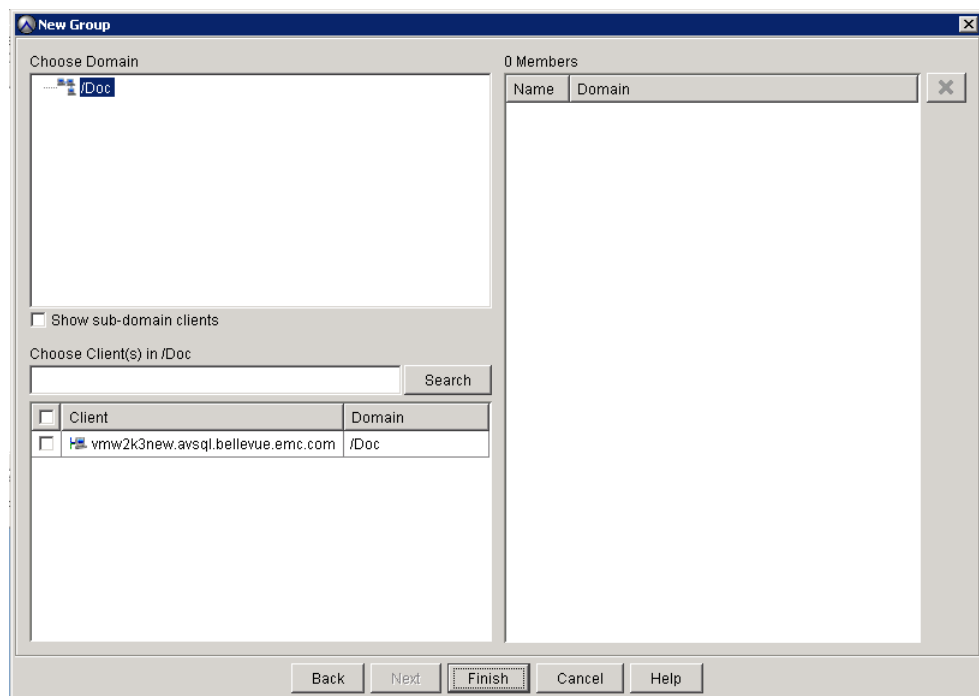
The screenshot shows a window titled "New Group" with a close button in the top right corner. Below the title bar, there is a dropdown menu labeled "Select an Existing Retention Policy:" with the value "/Default Retention". Below this is a section titled "Basic Retention Policy" containing three radio button options: "Retention period" (selected), "End date", and "No end date". The "Retention period" option has a text box containing "60" and a dropdown menu showing "days". The "End date" option has a dropdown menu showing "Mon 2013-02-11". Below these options is a checkbox labeled "Override basic retention policy for scheduled backups" which is checked, and a button labeled "Advanced...". At the bottom of the window, there are five buttons: "Back", "Next" (highlighted with a dashed border), "Finish", "Cancel", and "Help".

14. From the **Select An Existing Retention Policy** list, select a retention policy for the group.

You cannot edit retention policies on this page. The *EMC Avamar Administration Guide* provides information about editing retention policy properties.

15. Click **Next**.

The final **New Group** wizard page appears with a tree of domains and clients.



16. In the **Choose Domain** list, select the domain for the SQL Server client.

A list of clients in the domain appears in the lower-left list.

17. Select the checkbox next to the SQL Server client.

If you are backing up databases in an AlwaysOn availability group, then select the cluster client for the availability group listener.

If you are backing up databases on shared storage in a failover cluster, then select the cluster client for the virtual server.

18. (Optional) Select the checkboxes next to other clients to include in the group.

19. Click **Finish**.

Enabling scheduled backups

To ensure that the group is enabled for scheduled SQL Server backups:

1. In Avamar Administrator, click the **Policy** tab.

The **Policy** window appears.

2. On the **Policy Management** tab, select the **Groups** tab.

3. Select the group that you created in [“Creating a group” on page 74](#).

4. Select **Actions > Group > Disable Group** to clear the **Disable Group** option on the **Actions > Group** menu.

A confirmation message appears.

5. Click **Yes** to enable this group.

Setting backup options

You set plug-in options for an on-demand backup using the **Backup Command Line Options** dialog box. For a scheduled backup, you set plug-in options on the **Options** tab of the **New Dataset** dialog box.

To view all plug-in options, including advanced options, select the **Show Advanced Options** checkbox. The advanced options appear in red.

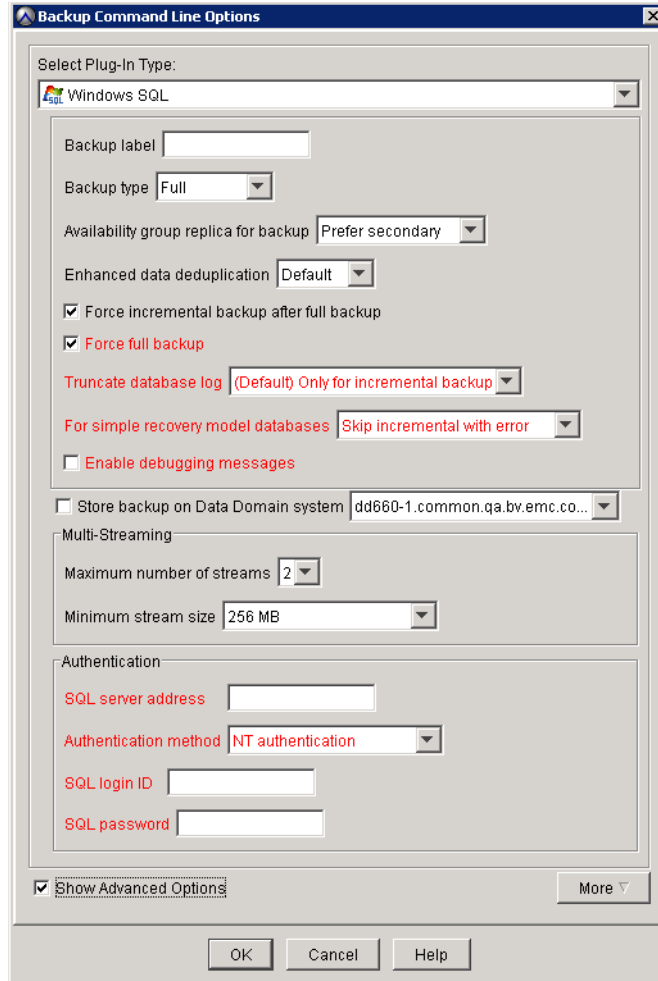


Figure 9 Backup Command Line Options dialog box

Backup label option

The **Backup label** option enables you to assign a descriptive label to the backup. This label appears in Avamar Administrator for the backup when you browse for backups to manage or restore.

Backup type options

The **Backup type**, **Force incremental backup after full backup**, and **Force full backup** options enable you to control the type of backup that occurs.

Backup type option

The **Backup type** option sets the backup level. There are three backup levels.

Full

Full backups include the entire database, including all objects, system tables, and data. As the backup operation progresses, it copies the transaction logs. The timing of the transaction log backup ensures that you can recover the complete database to the state it was in when the backup finished.

Differential

Differential backups include data that changed after the last full backup. The backup operation also copies relevant portions of the transaction logs.

When you select a differential backup to restore a database, the restore process restores the last full backup, followed by the differential backups performed after the full backup. Because a differential backup only saves changes to data, the backup is smaller and faster than a full backup. Therefore, you can perform a differential backup more often than a full backup.

Differential backups are used primarily in high volume systems where you must bring a failed database back online as quickly as possible. Differential backups are smaller than full backups, and so have less effect on the system where they run.

Incremental

By default, incremental, or transaction log, backups only back up the transaction logs. Transaction logs are serial records of all database modifications. The logs are used in recovery operations to update the database with complete transactions and roll back incomplete transactions. When you back up a transaction log, the backup stores all changes after the last transaction log backup. Transaction log backups record the state of the transaction log at the start of the backup (unlike full and differential backups, which record the state of the data at the end of the backup).

When a transaction log backup is complete, the log truncates to remove any transactions committed to the database. When restoring the database, you restore the data to the state it was in at the end of the last full or differential backup, and then sequentially restore the transaction log backups in order.

Considerations for selecting the backup type

Review the following considerations when you select the backup type:

- ◆ If the database uses the simple recovery model, then you cannot perform a transaction log backup of the database. Databases with the simple recovery model do not support transaction log backups. Databases that use the simple recovery model include system databases, such as the master and msdb databases.

If a transaction log backup includes databases with the simple recovery model and databases with other recovery models, then you can select how Avamar handles the databases with the simple recovery model. [“Simple recovery model databases option” on page 84](#) provides details.

- ◆ Microsoft SQL Server does not allow differential or transaction log (incremental) backups of the master database. You can only create full backups of the master database. If you try to perform a differential or transaction log backup of the master database, a full backup occurs instead.
- ◆ If you change the recovery model of a database, perform a full backup before you try to perform a differential or transaction log backup.
- ◆ You can perform full, differential, and transaction log backups on the primary replica in an AlwaysOn availability group. You can perform only full and transaction log backups on a secondary replica. If you try to perform a differential backup on a secondary replica, then the backup fails with an error message.

Force incremental backup after full backup option

If a full backup exists and then another full backup occurs, this option creates a transaction log (incremental) backup. The transaction log backup contains transactions from the end of the first full backup or the most recent transaction log backup, whichever is later, until the time that the forced transaction log backup occurs (after the first backup of the new backup cycle). This ensures that a point-in-time recovery to a point in time between the two full backups can occur, if required.

Three backup sessions occur when a forced transaction log backup occurs:

- ◆ The first session contains transactions that occur from the end of the first full backup until the start time of the forced transaction log backup.
- ◆ The second session contains transactions that occur from the end of the second full backup until the time that the forced transaction log backup occurs.
- ◆ The third session contains only the full backup.

To restore and recover the database to a point after the forced transaction log backup, select the last backup from the last backup session that has the most recent full backup, during the restore process.

Force full backup option

When you select the **Force full backup** option, a full backup occurs in the following scenarios:

- ◆ If you try to perform a differential or transaction log (incremental) backup and a full backup does not exist on the server.
- ◆ If you try to perform a differential or transaction log backup after you restore a database.
- ◆ If you try to perform a differential or transaction log backup after you restore the msdb database.
- ◆ If the backup process detects log gaps.

Effectively, this option automates taking a full backup when necessary.

As a best practice, perform a transaction log backup after the forced full backup.

NOTICE

If you perform transaction log and differential backups, EMC strongly recommends that you leave this option selected (the default setting) for all backups. Otherwise, you might not be able to restore data if there are no full backups on the Avamar server.

AlwaysOn availability group option

The **Availability group replica for backup** option controls the replica on which the backup occurs in an AlwaysOn environment. The following table describes each of the available values.

Table 5 Availability group replica for backup option values

Value	Description
Primary	Specifies that the backup should occur on the primary replica.
Prefer secondary	Specifies that the backup should occur on a secondary replica except when the primary replica is the only available replica or when none of the secondary replicas meet the requirements for backup. This is the default value.
Secondary only	Specifies that the backup should occur on the secondary replica. If no secondary replica is available, then the backup does not occur.
SQL Server defined	Specifies that the backup should use the backup preference specified in SQL Server settings.

You can increase performance of the primary replica when you perform backups on a secondary replica. However, the secondary replica must meet the requirements in [“AlwaysOn availability group requirements” on page 43](#).

You can perform only full and transaction log backups on a secondary replica. If you try to perform a differential backup on a secondary replica, then the backup fails with an error message.

Enhanced data deduplication option

The **Enhanced data deduplication** option controls the type of compression for data in a backup by setting the `--compress` option in `avtar`. Enhanced data deduplication typically reduces the amount of client data that the backup process must send to the server, but requires additional client CPU resources. The following table describes each of the available values.

Table 6 Enhanced data deduplication option values

Value	Description
Default	The <code>avtar</code> process evaluates some of the data in the backup to determine whether the data supports compression. If the evaluated data supports compression, then <code>avtar</code> compresses all data in the backup. If the evaluated data does not support compression, then <code>avtar</code> does not compress any data.
Disabled	The <code>avtar</code> process does not compress data during the backup.
Enabled	Forces <code>avtar</code> to compress data during the backup, even if the data does not support compression. Select this value only when you are certain that data in the backup supports compression. Otherwise, <code>avtar</code> unnecessarily uses additional client CPU resources to try to perform compression on data that does not support compression.

Truncate database log option

The **Truncate database log** option controls database transaction log truncation behavior. The following table describes each of the available values.

Table 7 Truncate database log option values

Value	Description
(Default) Only for incremental backup	Avamar truncates the database transaction log if the backup type is incremental (transaction log). No log truncation occurs if the backup type is full or differential.
For all backup types	Avamar truncates the database transaction log regardless of the backup type. This setting breaks the chain of log backups. Do not use this setting unless the backup type is full. Note: This value is not supported for SQL Server 2008 R2 and SQL Server 2012 environments.
Never	Avamar does not truncate the database transaction log under any circumstances.

Simple recovery model databases option

The **For simple recovery model databases** option specifies what Avamar does when you try to perform incremental (transaction log) backups of databases that use the simple recovery model, which does not support transaction log backups. The following table describes each of the available values.

Table 8 For simple recovery model databases option values

Value	Description
Skip incremental with error	If you select databases with different recovery models for the backup, then the backup does not include the databases with the simple recovery model. The backup completes with exceptions, and an error message is written to the log. If you select only databases with the simple recovery model for the backup, then the backup fails.
Skip incremental with warning	If you select databases with different recovery models for the backup, then the backup does not include databases with the simple recovery model. The backup completes successfully, and a warning is written to the log for each database that uses the simple recovery model. If you select only databases with the simple recovery model for the backup, then the backup fails.
Promote incremental to full	A full backup occurs automatically instead of a transaction log backup for databases that use the simple recovery model.

Enable debugging messages

When you select the **Enable debugging messages** option, maximum information is written to log files during the backup. If you select the checkbox, the backup process generates very large log files. Use this option only for debugging purposes.

Store backup on Data Domain system option

When you select the **Store backup on Data Domain system** option, Avamar stores the backup on a configured Data Domain system instead of on the Avamar server. [“Data Domain system support” on page 26](#) provides details on how Avamar works with a Data Domain system.

To store the backup on a Data Domain system, select the checkbox and then select the Data Domain system from the list.

Changing the backup storage

You can store backups on either the Avamar server or a Data Domain system. However, the full backup for a client and all subsequent transaction log (incremental) and differential backups must be stored on either the Avamar server or a single Data Domain system. Avamar does not support the following scenarios:

- ◆ Full backup on a Data Domain system, and transaction log or differential backups on the Avamar server
- ◆ Full backup on the Avamar server, and transaction log or differential backups on a Data Domain system
- ◆ Full backup on one Data Domain system, and transaction log or differential backups on another Data Domain system

As a result, if you change the server on which backups for a client are stored, then you must perform a full backup before any further transaction log or differential backups.

If you change the backup storage from one Data Domain system to another Data Domain system, you can restore to a point in time between the two full backups. Select the **Force incremental backup** checkbox when you perform the first full backup to the new Data Domain system. Otherwise, clear the **Force incremental backup** checkbox.

When you select the **Force incremental backup** checkbox, a restore error appears because full backup on one Data Domain system and incremental (transaction log) backup on another Data Domain system is not supported. As a result, the backup process creates a single backup set that includes both the new full backup and the forced incremental backup.

[“Requirements for point-in-time restore after changing the backup storage” on page 97](#) provides more information on restoring to the point in time between the two full backups.

Multi-streaming options

Multi-streaming enables you to improve backup and restore performance by backing up and restoring SQL Server data using multiple parallel data streams. The number of streams that you use to back up a database is automatically the same for the restore.

When you use the Avamar Plug-in for SQL Server, Avamar sends backup and restore data in multiple streams by spawning multiple instances of `avtar`. There is one `avtar` instance for each data stream, plus an additional `avtar` progress instance.

You can specify a maximum of 10 streams for each backup. [“Specifying the maximum number of streams” on page 87](#) provides guidance for how to select the maximum number of streams for a backup.

You can either back up multiple databases in parallel with one stream per database, or back up a single database using multiple parallel streams.

If you back up a single database with multiple parallel streams, then you can specify the minimum size of each stream during the backup. An equal amount of backup data is sent through each stream. [“Specifying the minimum stream size” on page 88](#) provides guidance for how to select the minimum stream size.

If you use multiple data streams to send backup data for a single database to the Avamar server or Data Domain system, then the backup for the database is stored as multiple files. If you restore the multi-stream backup to files, then the file name for each file is composed of the backup type and the stream number:

- ◆ f-0 indicates a full backup
- ◆ d-*n* indicates a differential backup
- ◆ i-*n* indicates a transaction log (incremental) backup

where *n* is the sequential number of the differential or incremental backup since the preceding full backup.

For example, if there are two files, `f-0.stream0` and `f-0.stream1`, then the backup files are for a full backup that was sent using two streams.

[“Restoring a database with SQL Server tools” on page 126](#) provides steps to use SQL Server management tools to restore a database from multiple files.

When to use multi-streaming

Multi-streaming is most effective at improving backup and restore performance on a computer that is running SQL Server with significant system resources, such as:

- ◆ Fast, multiprocessor CPU
- ◆ Large amounts of RAM
- ◆ Fast disk input/output (I/O)
- ◆ Fast network resources

If you use multi-streaming on a server with limited system resources, then the server may become unresponsive because of 100% resource usage during a backup or restore.

To improve disk I/O for backup and restore with multi-streaming, store databases on different physical drives. This enables Avamar to read and write data from each drive simultaneously during the backup or restore process.

Another way to improve disk I/O is to store databases on a high performance storage area network (SAN) attached to the computer that is running SQL Server, instead of on the local drives.

Calculating the number of streams for a database

You can either back up multiple databases in parallel with one stream per database, or back up a single database using multiple parallel streams. If you back up a single database with multiple parallel streams, then you can specify the minimum size of each stream during the backup.

After you determine the minimum stream size, you can calculate the number of streams used to back up the database by using the following equation:

$$\text{Database size} / \text{minimum stream size} = \text{Number of streams}$$

For example, if a database is 1,280 MB and the minimum stream size is the default setting of 256 MB, then the number of streams for a full backup of the database is five:

$$1,280 \text{ MB} / 256 = 5$$

For transaction log and differential backups, the size of the data to back up, and not the total database size, is used to calculate the number of streams.

If the database size is less than the minimum stream size, then Avamar uses a single stream to back up the database.

If you calculate the number of streams for a database based on the minimum stream size, and the number exceeds the maximum set for the backup, then the backup uses only the maximum number of streams.

Understanding the order of databases in a backup

When there are multiple databases in a backup, Avamar sets the order of the databases in the backup based on database size, with the largest database backed up first. Avamar calculates the number of streams to use to back up the database, and then allocates those streams for the backup.

If additional streams are available based on the specified maximum number of streams, then Avamar can use those streams to back up another database while the backup of the first database proceeds. The next database that is backed up depends on the size of the database and the number of streams required. Consider the following example:

1. There are four databases in the backup dataset:
 - The first and largest database requires four streams.
 - The second database requires three streams.
 - The third and fourth databases require two streams each.
2. The maximum number of streams for the backup is six.
3. When the backup starts, four of the six streams are allocated to the backup of the first database.
4. The second database requires three streams, but only two streams are available. As a result the second database is skipped.
5. The remaining two streams are allocated to the third database. The backup of the first database and the third database occur concurrently, using all six available streams.
6. When the backup of the first database completes, four streams become available. Three of the backup streams are allocated to the backup of the second database, leaving one stream available and unused.
7. The fourth database is backed up when either the backup of the third database completes or when the backup of the second database completes, depending on which completes first and two streams become available.

Specifying the maximum number of streams

You can specify a maximum of between one and 10 streams for each backup using the **Maximum number of streams** list on the **Backup Command Line Options** dialog box or on the **Options** tab of the **New Dataset** dialog box.

You cannot specify the maximum number of streams for a restore. The number of streams for the backup is the same for the restore.

In environments with significant system resources, as described in [“When to use multi-streaming” on page 86](#), specify the maximum of 10 streams to achieve the best possible performance.

In environments with fewer or limited system resources, specify a lower maximum number of streams, or disable multi-streaming altogether by specifying a maximum of one stream.

The optimal number of streams depends on the environment. In general, start with the highest maximum number of streams that you believe the environment can support. If resource usage during a backup is excessive and adversely impacts other processes on the server, then consider reducing the maximum number of streams.

Specifying the minimum stream size

You specify the minimum stream size using the **Minimum stream size** list on the **Backup Command Line Options** dialog box during an on-demand backup or on the **Options** tab when you create a dataset for a scheduled backup.

The minimum size of a stream must be one of the following values:

- ◆ One stream per database
- ◆ 256 MB (default)
- ◆ 512 MB
- ◆ 1,024 MB
- ◆ 2,048 MB

If the database size is less than the minimum stream size, then Avamar uses a single stream to back up the database.

Specifying the minimum stream size enables you to control how Avamar allocates the available streams to back up the databases in a backup set. [“Calculating the number of streams for a database” on page 86](#) and [“Understanding the order of databases in a backup” on page 87](#) explain how Avamar allocates streams to back up databases based on the maximum number of streams and the minimum stream size.

The optimal value for the minimum stream size depends on the following criteria:

- ◆ Number of databases
- ◆ Amount of data:
 - Consider the size of the database for full backups.
 - Estimate the amount of changed data for differential backups.
 - Consider the log size for transaction log backups.
- ◆ Whether databases are on a single drive or on multiple drives
- ◆ Available system resources
- ◆ Restore requirements

The following table provides recommendations for setting the minimum stream size depending on the backup set and environment. The information in the following table is intended only as general guidelines. The optimal setting for minimum stream size for an environment depends on many factors.

NOTICE

The following table provides recommendations for the specified backup set and environment for a full backup, which includes all data in the database. The recommended minimum stream size is likely to be different for differential and transaction log backups. To determine the optimal minimum stream size, estimate the amount of changed data for differential backups and the size of the logs for transaction log backups.

Table 9 Recommended minimum stream size by backup set and environment (page 1 of 2)

Backup set and environment	Recommended minimum stream size
One large (more than 12 GB) database on a single drive	<p>Select any of the following values:</p> <ul style="list-style-type: none"> • 256 MB (default) • 512 MB • 1,024 MB • 2,048 MB <p>The database uses all available streams during the backup, and the minimum stream size does not impact performance. Backup performance could be limited by disk I/O.</p>
A few large (more than 12 GB) databases on a single drive	<p>Select any of the following values:</p> <ul style="list-style-type: none"> • 256 MB (default) • 512 MB • 1,024 MB • 2,048 MB <p>Avamar backs up each database sequentially but with multiple streams. Because the databases are large, each database uses all available streams during the backup, regardless of the minimum stream size. In addition, the minimum stream size does not impact performance.</p>
A few large (more than 12 GB) databases of approximately the same size, each on a separate drive	<p>Select One stream per database. This approach backs up multiple databases in parallel and maximizes disk I/O potential by reading from each drive simultaneously.</p>

Table 9 Recommended minimum stream size by backup set and environment (page 2 of 2)

Backup set and environment	Recommended minimum stream size
Mixed environment with both small and large databases either on a single drive or on multiple drives	<p>For each database in the backup set, divide the database size (or the amount of changed data or the log size for differential and transaction log backups, respectively) by each minimum stream size value (256 MB; 512 MB; 1,024 MB; and 2,048 MB) to determine how many streams that Avamar would use to back up the database. Consider the maximum number of streams for the backup and the order in which Avamar would back up the databases. Evaluate the optimal minimum stream size based on these factors.</p> <p>“Examples of minimum stream size in a mixed database environment” on page 91 provides examples of how different minimum stream sizes impact system performance in a mixed environment.</p>
Mixed environment with a few large (more than 12 GB) databases, and one or two very large (more than 500 GB) databases, either on a single drive or on multiple drives	<p>For each database in the backup set, divide the database size (or the amount of changed data or the log size for differential and transaction log backups, respectively) by each minimum stream size value (256 MB; 512 MB; 1,024 MB; and 2,048 MB) to determine how many streams that Avamar would use to back up the database. Consider the maximum number of streams for the backup and the order in which Avamar would back up the databases. Evaluate the optimal minimum stream size based on these factors.</p> <p>“Examples of minimum stream size in a mixed database environment” on page 91 provides examples of how different minimum stream sizes impact system performance in a mixed environment.</p>
Many small databases either on a single drive or on multiple drives	<p>Select One stream per database. This approach backs up multiple databases in parallel, with one stream for each database.</p>

Examples of minimum stream size in a mixed database environment

The following table provides examples of how different minimum stream sizes impact stream allocation for environments with mixed database sizes.

NOTICE

The examples in the following table are for full backups, which include all data in the database. The result is likely to be different for differential backups, which include only changed data, and transaction log backups, which include only the logs.

Table 10 Examples of minimum stream size in mixed environments (page 1 of 2)

Database environment	Maximum number of streams	Minimum stream size	Result
One database of 9 GB and 100 databases of 100 MB	10	256 MB	Avamar uses all 10 streams to back up the 9 GB database. When the backup of the 9 GB database completes, Avamar uses all 10 streams to back up the 100 MB databases, with one stream for each database.
	10	2,048 MB	Avamar uses four streams to back up the 9 GB database, and the remaining six to back up six of the 100 MB databases. When the backup of the 9 GB database completes, Avamar uses all 10 streams to back up the remaining 100 MB databases, with one stream for each database.
Several large databases of more than 12 GB and many small databases of 2 GB or less	10	2,048 MB	Avamar uses all 10 streams to back up the large databases, one database at a time, in order from largest to smallest. When the backup of the large databases completes, Avamar uses all 10 streams to back up the small databases, with one stream for each database, in order from the largest database to the smallest database.

Table 10 Examples of minimum stream size in mixed environments (page 2 of 2)

Database environment	Maximum number of streams	Minimum stream size	Result
Four large (more than 12 GB) databases, and two very large (more than 500 GB) databases	6	256 MB	Avamar uses all six streams to back up each database, one database at a time, in order from largest to smallest.
	6	One stream per database	Avamar uses one stream to back up each database. When the backup starts, Avamar uses all six streams—one for each database. As the backup continues and completes for the four large databases, Avamar uses only two streams—one for each of the two very large databases. Four of the six available streams remain idle. Therefore, backup performance in this case is likely to be slower than when you select a minimum stream size of 256 MB.

Considering restore requirements when setting the minimum stream size

As you evaluate the optimal minimum stream size for an environment, remember that the multi-streaming settings that you specify for the backup are also used for the restore. If restore performance is an issue, then evaluate whether the minimum stream size used for the backup is also optimal for restore.

For example, a minimum stream size of one stream per database is recommended for backups in environments with a few large (more than 12 GB) databases of approximately the same size, each on a separate drive. However, when you restore a single database from that backup, the restore process uses only one stream.

Authentication options

Authentication options enable you to specify whether Avamar uses Windows (NT) authentication or SQL Server authentication to connect to SQL Server during a backup.

If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account. [“Security requirements” on page 40](#) provides more information.

To specify authentication options:

1. In the **Backup Command Line Options** dialog box or on the **Options** tab of the **New Dataset** dialog box, select the **Show Advanced Options** checkbox.
Several advanced options, including the authentication options, appear in red.
2. In the **SQL server address** box, specify the hostname or IP address of the server to connect to.
3. From the **Authentication method** list, select whether to use **NT authentication** or **SQL Server authentication**.

- If you select SQL Server authentication, specify the login ID and password for the SQL Server account in the **SQL login ID** and **SQL password** boxes, respectively.

Monitoring backups

To monitor the status of on-demand or scheduled backups:

- In Avamar Administrator, click the **Activity** tab.

The **Activity** window appears.

- Select the **Activity Monitor** tab.

A list of all activities appears.

Status	Error Code	Start Time (PST)	Elapsed	End Time (PST)	Type	Server	Progress Bytes	New Bytes	
Completed		2012-12-10 19:27	00h:00m:29s	2012-12-10 19:28	On-Demand Backup	Avamar	4.7 MB	2.4%	agcls2l
Completed		2012-12-12 10:07	00h:00m:58s	2012-12-12 10:08	Restore	Avamar	5.6 MB	11.4%	agnew
Completed		2012-12-11 18:51	00h:00m:21s	2012-12-11 18:52	On-Demand Backup	Avamar	18.8 MB	3%	vm2k6r
Failed	10007	2012-12-12 15:25	00h:00m:08s	2012-12-12 15:25	On-Demand Backup	Avamar	0	0%	agcls2l
Completed		2012-12-11 23:53	00h:00m:26s	2012-12-11 23:53	On-Demand Backup	Avamar	< 1MB	4.4%	agcls2l
Completed		2012-12-12 08:51	00h:00m:12s	2012-12-12 08:51	Restore	Avamar	4.7 MB	4.2%	w2k3x
Completed		2012-12-12 11:37	00h:00m:21s	2012-12-12 11:38	On-Demand Backup	Avamar	4.9 MB	4%	agcls2l
Completed		2012-12-10 13:58	00h:00m:28s	2012-12-10 13:58	On-Demand Backup	Avamar	4.7 MB	1.4%	w2k3x
Completed		2012-12-12 14:52	00h:00m:47s	2012-12-12 14:53	Restore	Avamar	4.7 MB	9.8%	agcls2l
Completed		2012-12-10 13:22	00h:00m:28s	2012-12-10 13:23	Restore	Avamar	4.7 MB	5.1%	w2k3x
Completed w/Exception(s)	10020	2012-12-10 16:23	00h:00m:27s	2012-12-10 16:24	On-Demand Backup	Avamar	4.7 MB	1.1%	w2k3x
Failed	10007	2012-12-12 19:50	00h:00m:08s	2012-12-12 19:50	On-Demand Backup	Avamar	0	0%	agcls2l
Completed		2012-12-11 23:32	00h:00m:19s	2012-12-11 23:33	On-Demand Backup	Avamar	4.7 MB	1%	agcls2l
Completed		2012-12-12 13:25	00h:00m:31s	2012-12-12 13:25	On-Demand Backup	Avamar	4.9 MB	2.1%	agcls2l
Completed w/Exception(s)	10020	2012-12-10 16:23	00h:00m:27s	2012-12-10 16:24	On-Demand Backup	Avamar	4.7 MB	1.1%	w2k3x

- To filter the results to display only backup activity, select **Actions** > **Filter**.

The **Filter Activity** dialog box appears.

- In the **Filter Activity** dialog box, select **All Backups** from the **Type** list.

- Click **OK**.

Troubleshooting backups

The following topics provide details on how to troubleshoot problems that may occur during backups with the Avamar Plug-in for SQL Server.

Error appears for a forced incremental backup after a restore

An error appears in the Avamar client log file if you perform a full backup with a forced incremental backup after you restore a database. The full backup completes successfully, but Avamar does not perform the incremental backup.

A forced incremental backup contains transactions that occur from the end of the first full backup or the most recent transaction log backup, whichever is later, until the forced incremental backup occurs (after the first backup of the new backup cycle). This ensures that a point-in-time recovery to a point in time between the two full backups can occur, if required.

If a restore occurs between the two full backups, then the log chain breaks. Avamar cannot create a transaction log backup (forced incremental backup) for the transactions between the backups.

Canceling backups

To cancel a backup that is in progress:

1. In Avamar Administrator, click the **Activity** tab.

The **Activity** window appears.

2. Select the **Activity Monitor** tab.

A list of all activities appears.

3. From the list of activities, click the backup to highlight it.

4. Select **Actions > Cancel Activity**.

A confirmation message appears.

5. Click **Yes**.

The cancellation may take five minutes or more to complete. The backup may complete before the cancellation finishes.

6. Perform a new full backup immediately.

When you cancel a backup, it breaks the log/backup chain. As a result, when you restore the backup, the database cannot recover to the specified point in time and may remain in a restoring state.

If the issue occurs and there is a valid incremental (transaction log) backup, then you can restore the database to an operating system file. Then, use SQL Server tools to restore the database to a point in time.

CHAPTER 4

Restore

The following topics explain how to restore SQL Server data by using the Avamar Plug-in for SQL Server:

- ◆ Restore requirements 96
- ◆ Finding a backup..... 98
- ◆ Restoring to the original location..... 106
- ◆ Restoring to a new database in the original instance 109
- ◆ Restoring to a different instance 113
- ◆ Restoring to a file 119
- ◆ Restoring system databases..... 132
- ◆ Restoring to an AlwaysOn availability group 138
- ◆ Restoring a database with an intact log file 141
- ◆ Setting restore options..... 142
- ◆ Monitoring restores..... 149
- ◆ Canceling restores 150

Restore requirements

To ensure a successful restore, ensure that the environment meets the necessary requirements.

The Avamar Plug-in for SQL Server performs only offline restores of SQL Server data. Online restore is not supported.

Software requirements for restore

To restore an instance, database, filegroup, or file to SQL Server by using the Avamar Plug-in for SQL Server, ensure that the software in the environment meets the following requirements:

- ◆ The following software must be running on both the source and destination systems:
 - Microsoft SQL Server
 - Avamar Client for Windows
 - Avamar Plug-in for SQL Server

If the Avamar Plug-in for SQL Server is not installed on the target server, or you want to use the standard SQL Server restore tools for features that the Avamar Plug-in for SQL Server does not provide, then you can restore a SQL Server database from an Avamar backup to operating system files. You can then use SQL Server tools to restore the database.

- ◆ The target SQL Server installation for a restore must have the same SQL Server version and service pack as the SQL Server installation on which the backup occurred. Otherwise, restore fails.
- ◆ The destination server must be registered with the same Avamar server as the source.
- ◆ If the SQL Server installation is in a failover cluster, then you configured the Avamar cluster client for all SQL Server cluster nodes.
- ◆ If you enabled AlwaysOn availability groups, then you configured the Avamar cluster client for the availability group listener.

Tail-log backup and point-in-time restore requirements

To perform a tail-log backup during the restore process, the database must be online and using either the full or bulk-logged recovery model. To perform a point-in-time restore, the database must be using the full recovery model. As a result, you cannot perform either a tail-log backup or a point-in-time restore of system databases such as the master and msdb databases because those databases use the simple recovery model.

To restore to a specific point in time, you must provide the transaction date and time or named mark to which to recover from the SQL Server transaction log. The SQL Server documentation on the Microsoft website provides details on how to access transaction log information.

The point in time that you are restoring to must be after the finish time for the most recent full backup. In addition, if the point in time is before the start time of the most recent transaction log (incremental) backup, then a tail-log backup is not required. However, a tail-log backup *is* required if the point in time is after the most recent transaction log backup.

When you specify the point in time for restore, do not specify the start time of the selected transaction log backup if it is not the last backup in the backup sequence. Otherwise, the restore fails and a tail-log backup does not occur even if you select the **Tail-log backup** option.

Requirements for point-in-time restore after changing the backup storage

You can restore to a point in time between two full backups when you change the backup storage from one Data Domain system to another Data Domain system.

To restore to a point in time between a full backup on the first Data Domain system and the first full backup to the new Data Domain system, select the **Force incremental backup** checkbox when you perform the first full backup to the new Data Domain system. Otherwise, clear the **Force incremental backup** checkbox.

When you select the **Force incremental backup** checkbox, a restore error appears because full backup on one Data Domain system and incremental (transaction log) backup on another Data Domain system is not supported. As a result, the backup process creates a single backup set that includes both the new full backup and the forced incremental backup.

To restore the database to a point in time between the full backup on the first Data Domain system and the full backup on the second Data Domain system:

1. Restore the full backup from the Data Domain system to a file.
2. Restore the forced incremental (transaction log) backup from the second Data Domain system to a file.
3. Use those files to restore the database to the necessary point in time using SQL Server tools such as SQL Server Management Studio or the Microsoft SQL Server `sqlcmd` utility. The Microsoft website also provides full details on how to use SQL Server Management Studio to restore a database backup to a specific point in time.

[“Restoring to a file” on page 119](#) provides details on restoring to a file, and examples of using SQL Server tools to restore from a file to SQL Server.

Requirements to restore secondary database files

If you are restoring the secondary data files of a database to the original location, then the following restore requirements apply:

- ◆ You must select the most recent backup of the database for the restore.
- ◆ You must perform a tail-log backup.
- ◆ You must perform the restore with a single restore operation.

You cannot restore additional secondary data files after you perform a restore with a tail-log backup.

To restore the secondary data files of a database with two different restore operations, restore the backup to a file and then use SQL Server management tools to perform the restores.

Requirements to restore the Report Server database

Stop the SQL Server Reporting Services before you restore the Report Server database. Otherwise, the Report Server database does not restore correctly.

SQL Server write permission requirements

SQL Server must have write permissions to the location to which you restore data. Otherwise, the restore fails with an “Access is denied” error.

Finding a backup

You can find Avamar backups for restore by either:

- ◆ Date
- ◆ Instance or database

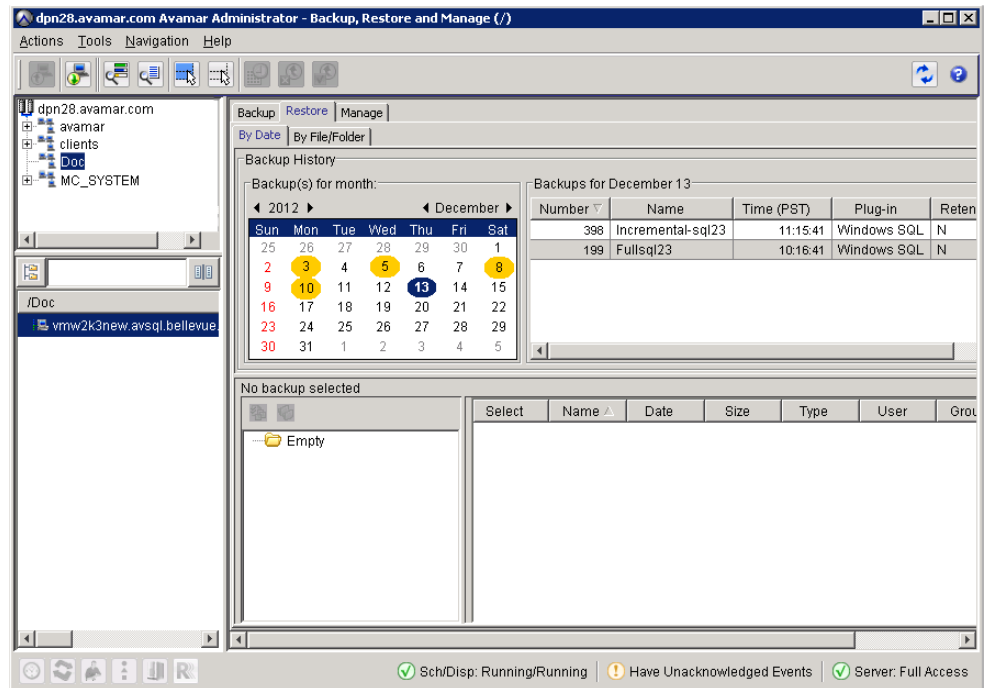
When to find a backup by date

Locate backups by date when:

- ◆ A single backup set includes all instances and databases for a client.
- ◆ The exact path or name of the instance or database that you want to restore is unknown.
- ◆ The content from a backup that you want to restore is before a specific date or event. For example, you know approximately when a database was lost or corrupted, and need to find the last backup before that date.
- ◆ The specific types of backups are known. For example, you run scheduled System State backups every Wednesday and Saturday night, and you run full volume backups daily. If you need to rebuild a server, you can select the System State backup with the date closest to the event that caused the loss of data.

How to find a backup by date

1. In Avamar Administrator, click the **Backup & Restore** launcher button.
The **Backup, Restore and Manage** window appears.
2. In the domain tree, select the domain for the client.
3. From the list of clients, select the computer that is running SQL Server. Keep the following points in mind when you select a client:
 - If you are restoring databases in an AlwaysOn availability group, then select the cluster client for the availability group listener.
 - If you are restoring databases on shared storage in a failover cluster, then select the cluster client for the virtual server.
 - You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
4. Click the **Restore** tab.

5. Click the **By Date** tab.

6. Select a backup from the calendar:

- a. Use the year and month navigational arrows to browse to a backup.

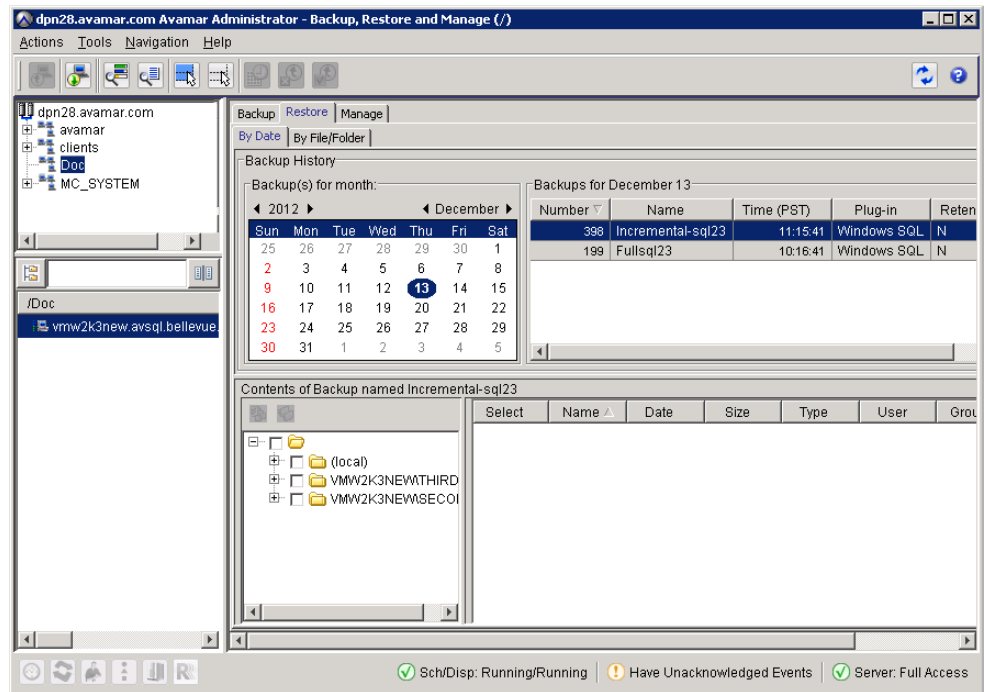
A yellow highlight on a date indicates that there was a valid backup on that date.

- b. Click a date highlighted by yellow.

A list of backups on that date appears in the **Backups** table next to the calendar.

If a backup is the second or later full backup and you selected the **Force incremental backup after full backup** checkbox for the backup, then three backups appear in the **Backups** table:

- The first backup in the list is the forced incremental backup. You can use this backup to restore from the full backup through the forced incremental backup.
- The backup with `forced_incremental` appended to the backup label is the forced incremental backup. You can use this backup to restore from the previous full backup through the forced incremental backup.
- The backup with `full_backup_only` appended to the backup label is the full backup.

7. Select the backup to restore from the **Backups** table.

Keep in mind the following points when you select the backup:

- In the **Type** column in the lower-right pane, f-0 indicates a full backup, d-*n* indicates a differential backup, and i-*n* indicates a transaction log (incremental) backup.
- If you are restoring from a transaction log or differential backup, select the backup that corresponds to the date and time to which you want to recover.

During the restore process, Avamar automatically restores any necessary data from the full backup, then restores and applies the intervening backup files as necessary. In other words, you do not need to select the full backup in addition to the transaction log or differential backup.

- Select the most recent backup if you plan to perform a tail-log backup and recover to a point in time since that last backup. A tail-log backup includes only transactions that have not been included yet in a backup.
8. In the two bottom panes of the **Backup, Restore and Manage** window, select the data to restore:
- To restore everything listed for the instance, select the checkbox next to the instance in the folder tree in the lower-left pane.
 - To restore a database and its logs, expand the node for the instance in the folder tree in the lower-left pane, and then select the checkbox next to the database.
 - To restore a filegroup, expand the node for the instance in the folder tree in the lower-left pane, select the database in the lower-left pane, and then select the checkbox next to the files in the filegroup in the lower-right pane.

If there are multiple files in the filegroup, ensure that you select the checkbox next to each file to ensure that you restore the entire filegroup.

The name of the filegroup to which a file belongs appears in the **Filegroup** column of the lower-right pane.

If you are restoring from a transaction log or differential backup and you plan to restore the files to a different instance, ensure that you select the checkbox next to all files in all filegroups. You cannot restore individual files from a transaction log or differential backup to a different instance.

Note: You cannot accurately determine the exact size of a restored database until after the restore operation completes. As a result, database sizes that appear in Avamar Administrator when you perform a restore may be smaller than when you perform a backup.

9. Continue with the restore as described in the following topics:
 - [“Restoring to the original location” on page 106](#)
 - [“Restoring to a new database in the original instance” on page 109](#)
 - [“Restoring to a different instance” on page 113](#)
 - [“Restoring to a file” on page 119](#)
 - [“Restoring system databases” on page 132](#)

When to find a backup for an instance or database

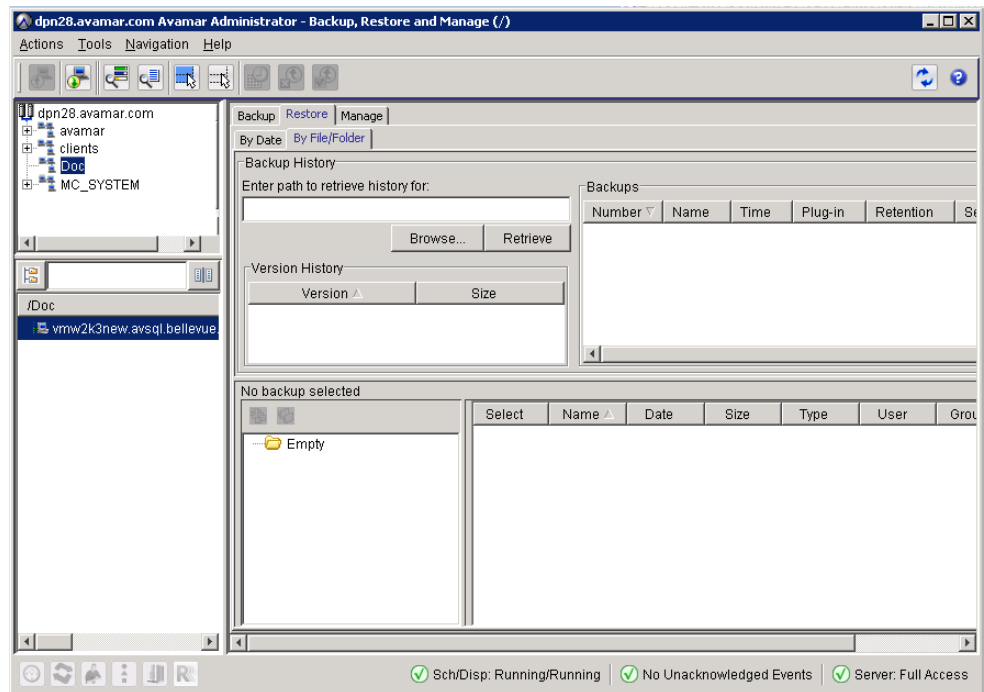
Locate backups by the instance or database in the backup when:

- ◆ Each instance or database is backed up in a separate backup set. For example, you know that `\\Server_name\Databases\Database_1` is backed up in one backup set and `\\Server_name\Databases\Database_2` is backed up in another backup set. If you know that the content you need is in `Database_2`, or is the entire `Database_2` database, then you can specify the path or browse to the `Database_2` folder.
- ◆ You want to see multiple versions of the same instance or database.
- ◆ The date of the backup or the content of a backup is unknown, but you know the name of the instance or database.

How to find a backup for an instance or database

To find a backup for an instance or database:

1. In Avamar Administrator, click the **Backup & Restore** launcher button.
The **Backup, Restore and Manage** window appears.
2. In the domain tree, select the domain for the client.
3. From the list of clients, select the computer that is running SQL Server. Keep the following points in mind when you select a client:
 - If you are restoring databases in an AlwaysOn availability group, then select the cluster client for the availability group listener.
 - If you are restoring databases on shared storage in a failover cluster, then select the cluster client for the virtual server.
 - You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
4. Click the **Restore** tab.
5. Click the **By File/Folder** tab.



6. In the **Enter path to retrieve history for** text box, specify the path to the instance or database by using one of the methods in the following table.

Table 11 Method and steps to browse to a backup by instance or database

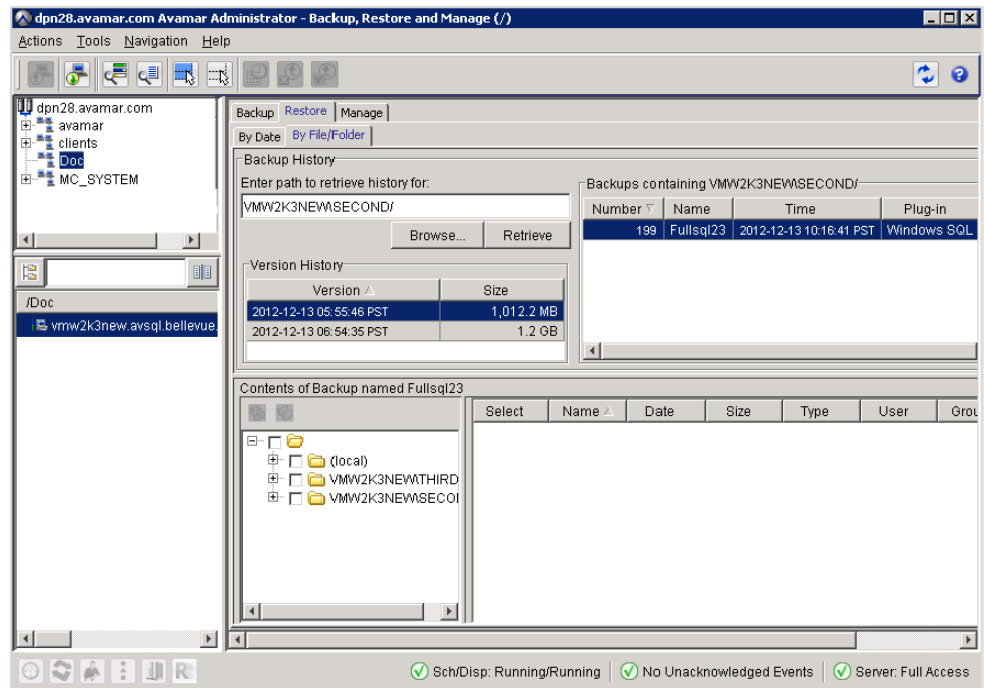
Method	Steps
Browse to the instance or database	<ol style="list-style-type: none"> 1. Click Browse. The Select File or Folder dialog box appears. 2. Select the client from the Clients tree in the left pane. 3. In the middle pane, expand the Windows SQL plug-in node. 4. Under the Windows SQL plug-in node, select the SQL Server instance that contains the databases for restore. A list of the databases for that instance appears in the right pane of the Select File or Folder dialog box. 5. To select all databases in an instance, select the checkbox next to the instance in the middle pane. Or, to select an individual database, select the checkbox next to the database in the right pane. 6. Click OK.
Type the path to the instance or database	<p>Type the full path to the client folder or file in the Enter path to retrieve history for text box using one of the following formats:</p> <ul style="list-style-type: none"> • To restore the local instance, type <code>(local)</code>. • To restore a database in the local instance, type <code>(local)/database/</code>. • To restore a named instance, type <code>client\instance/</code>. • To restore a database in a named instance, type <code>client\instance/database/</code>. • To restore a database if there is only one instance on the client and it is not the local instance, type <code>client/database/</code>. <p>where <i>client</i> is the name of the computer that is running SQL Server, <i>instance</i> is the name of the named instance, and <i>database</i> is the name of the database.</p>

7. Click **Retrieve**.

The **Version History** table lists all versions and sizes for that instance or database in backups from the selected client.

8. Select the instance or database version in the **Version History** table.

All backups for the selected client that contain the selected version appear in the **Backups** table next to the **Version History** table.

9. Select the backup to restore from the **Backups** table.

Keep in mind the following points when you select the backup:

- In the **Type** column in the lower-right pane, f-0 indicates a full backup, d-*n* indicates a differential backup, and i-*n* indicates a transaction log (incremental) backup.
- If you are restoring from a transaction log or differential backup, select the backup that corresponds to the date and time to which you want to recover.

During the restore process, Avamar automatically restores any necessary data from the full backup, then restores and applies the intervening backup files as necessary. In other words, you do not need to select the full backup in addition to the transaction log or differential backup.

- Select the most recent backup if you plan to perform a tail-log backup and recover to a point in time after that last backup. A tail-log backup includes only transactions that are not yet in a backup.

10. In the two bottom panes of the **Backup, Restore and Manage** window, select the data to restore:

- To restore everything listed for the instance, select the checkbox next to the instance in the folder tree in the lower-left pane.
- To restore a database and its logs, expand the node for the instance in the folder tree in the lower-left pane, and then select the checkbox next to the database.
- To restore a filegroup, expand the node for the instance in the folder tree in the lower-left pane, select the database in the lower-left pane, and then select the checkbox next to the files in the filegroup in the lower-right pane.

If there are multiple files in the filegroup, ensure that you select the checkbox next to each file to ensure that you restore the entire filegroup.

The name of the filegroup to which a file belongs appears in the **Filegroup** column of the lower-right pane.

If you are restoring from a transaction log or differential backup and you plan to restore the files to a different instance, ensure that you select the checkbox next to all files in all filegroups. You cannot restore individual files from a transaction log or differential backup to a different instance.

Note: You cannot accurately determine the exact size of a restored database until after the restore operation completes. As a result, database sizes that appear in Avamar Administrator when you perform a restore may be smaller than when you perform a backup.

11. Continue with the restore as described in the following topics:

- [“Restoring to the original location” on page 106](#)
- [“Restoring to a new database in the original instance” on page 109](#)
- [“Restoring to a different instance” on page 113](#)
- [“Restoring to a file” on page 119](#)
- [“Restoring system databases” on page 132](#)

Restoring to the original location

You can restore a SQL Server instance, database, filegroup, or file to its original location.

NOTICE

If you are restoring to the original location in an AlwaysOn availability group, use the instructions in [“Restoring to an AlwaysOn availability group” on page 138](#).

Options when restoring to the original location

There are two restore options when you restore a SQL Server instance, database, or filegroup to its original location:

- ◆ A standard restore with a tail-log backup is the most common restore procedure. During this procedure, a tail-log backup occurs to capture transactions that are not yet in a backup. Then the database is restored from the most recent full backup and any differential or transaction log backups.
- ◆ A restore with the SQL Server `REPLACE` option that completely overwrites the database might be necessary, for example, if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log:

```
One or more devices or files already exist.
Reissue the statement using the WITH REPLACE
option to overwrite these files and devices.
```

NOTICE

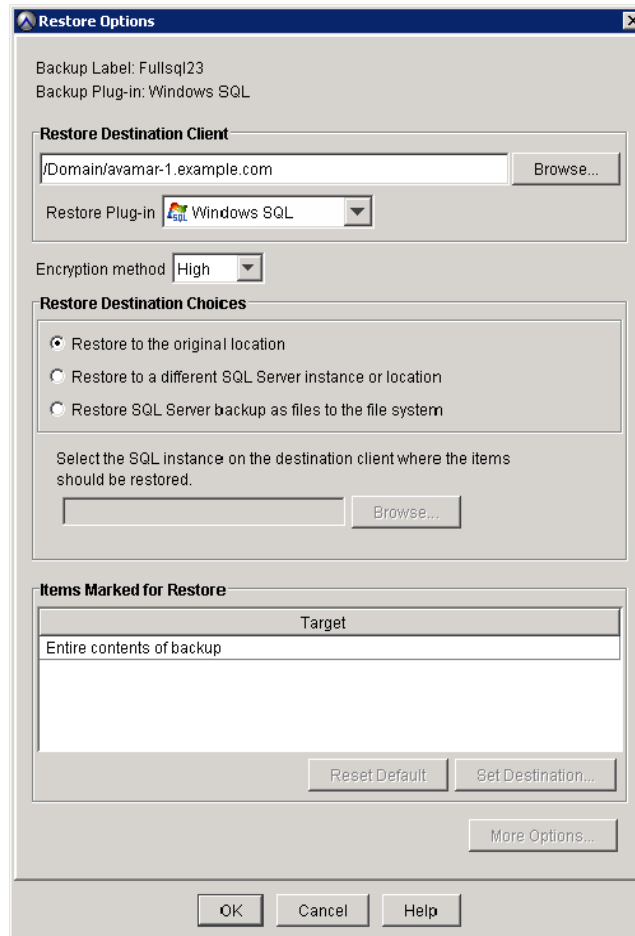
When you select the Avamar option to use the SQL Server `REPLACE` option, it adds an SQL `WITH REPLACE` clause statement to the restore Transact-SQL command. This statement overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The Microsoft Transact-SQL documentation provides more information in the `RESTORE` command section.

How to restore to the original location

1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 96](#).
2. In Avamar Administrator, click the **Backup & Restore** launcher button.
The **Backup, Restore and Manage** window appears.
3. Find the backup and select the instance, database, filegroup, or file to restore, as discussed in the following topics:
 - [“How to find a backup by date” on page 98](#)
 - [“How to find a backup for an instance or database” on page 102](#)

4. Select **Actions > Restore Now**.

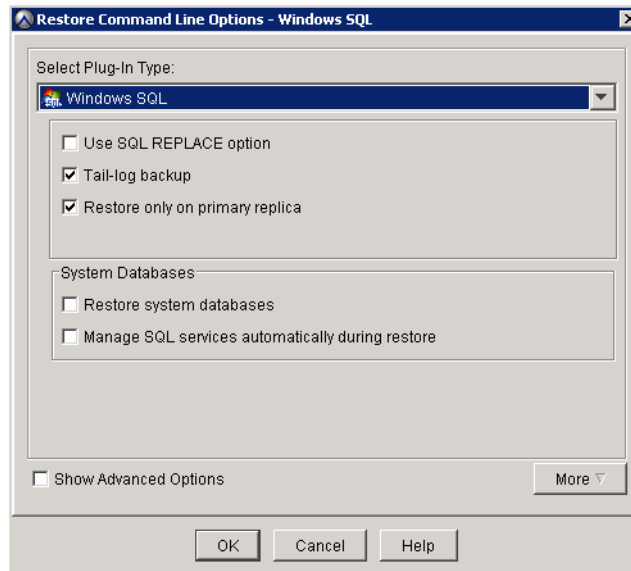
The **Restore Options** dialog box appears.



5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
6. Select the encryption method to use for client/server data transfer during the restore.
The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.
7. Leave the default selection of **Restore to the original location**.

8. Click **More Options**.

The **Restore Command Line Options - Windows SQL** dialog box appears.



9. Select the restore and recovery options:

- To create a tail-log backup and perform a direct restore with recovery, leave the **Use SQL REPLACE option** checkbox clear and the **Tail-log backup** checkbox selected.
- To perform a direct restore with `REPLACE`, select the **Use SQL REPLACE option** checkbox and clear the **Tail-log backup** checkbox.

10. If you are restoring in an AlwaysOn availability group and you want to restore on only the primary replica, select the **Restore only on primary replica** checkbox.

Leave the checkbox clear to automatically restore to both the primary replica and any secondary replicas.

[“Restoring to an AlwaysOn availability group” on page 138](#) and [“Restore only on primary replica” on page 144](#) provide details.

11. To restore system databases, set the plug-in options as discussed in [“System database restore options” on page 146](#).12. (Optional) Select **Show Advanced Options**, and set the advanced options as discussed in the following topics:

- [“Recovery operation options” on page 145](#)
- [“Authentication options” on page 147](#)
- [“Point-in-time recovery options” on page 148](#)

13. Disregard the redirected restore options, which are only necessary when you restore to a different location.

14. Click **OK** on the **Restore Command Line Options** dialog box.

15. Click **OK** on the **Restore Options** dialog box.

The **Restore Request** dialog box indicates that the restore started.

16. Click **Close**.

NOTICE

If you perform a tail-log backup and the tail-log backup fails to complete, then the restore cannot take place. Review the log file to determine the cause of the failure. Correct the problem, and then restart the restore. Keep in mind that if you clear the **Tail-log backup** checkbox to prevent the tail-log backup from occurring, then the restore includes only the transactions up to the selected backup. You may lose any transactions in the tail of the log.

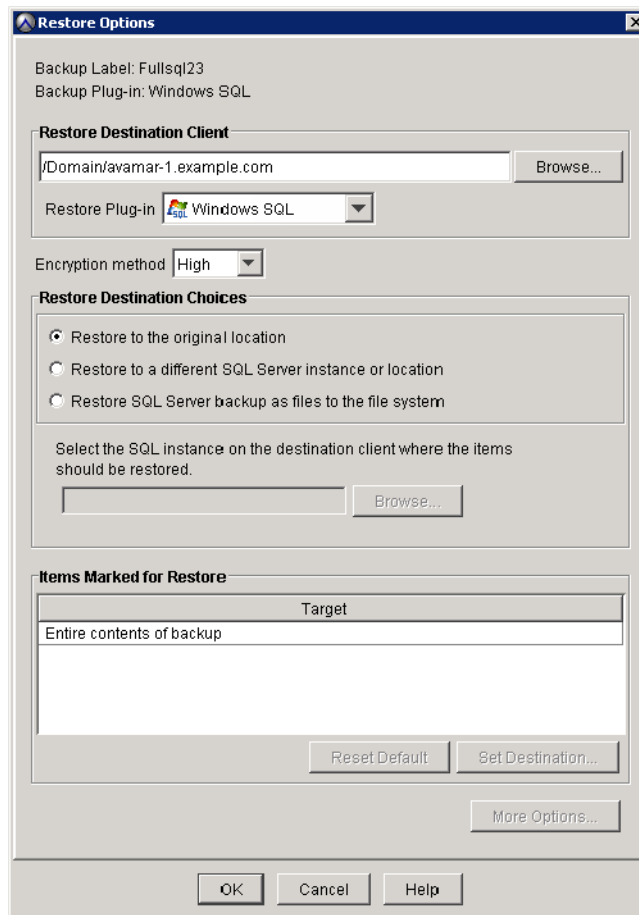
Restoring to a new database in the original instance

To restore a database to the original SQL Server instance on the same SQL Server client but with a new database name:

1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 96](#).
2. In Avamar Administrator, click the **Backup & Restore** launcher button.
The **Backup, Restore and Manage** window appears.
3. Find the backup and select the database to restore, as discussed in the following topics:
 - [“How to find a backup by date” on page 98](#)
 - [“How to find a backup for an instance or database” on page 102](#)

4. Select **Actions > Restore Now**.

The **Restore Options** dialog box appears.



5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.

6. Select the encryption method to use for client/server data transfer during the restore.

The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

7. Leave the default selection of **Restore to the original location**.

8. Click **More Options**.

The **Restore Command Line Options - Windows SQL** dialog box appears.

9. Select the **Show Advanced Options** checkbox.

Several advanced options appear in red.

The screenshot shows the 'Restore Command Line Options - Windows SQL' dialog box. The 'Show Advanced Options' checkbox is checked. The following options are highlighted in red:

- Use SQL REPLACE option
- Tail-log backup
- Restore only on primary replica
- Recovery operation: RECOVERY
- Standby file location: [text box]
- Enable debugging messages
- System Databases:
 - Restore system databases
 - Manage SQL services automatically during restore
- Redirected Restore:
 - New database name: [text box]
 - Alternate database location: [text box]
 - Alternate log location: Same as alternate database location
 - Path to alternate log location: [text box]
- Authentication:
 - SQL server address: [text box]
 - Authentication method: NT authentication
 - SQL login ID: [text box]
 - SQL password: [text box]
- Point-in-time Recovery:
 - Point-in-time recovery mode: None
 - Point-in-time or mark name string: [text box]
 - Mark recovery point: At mark
 - Mark is after date/time: [text box]

At the bottom, the 'Show Advanced Options' checkbox is checked, and there is a 'More' button. The 'OK', 'Cancel', and 'Help' buttons are at the very bottom.

10. Leave the **Use SQL REPLACE option** checkbox clear.
11. Specify whether to perform a tail-log backup during the restore by selecting or clearing the **Tail-log backup** checkbox. [“Tail-log backup” on page 143](#) provides details.
12. If you are restoring in an AlwaysOn availability group and you want to restore the database with the new name on only the primary replica, select the **Restore only on primary replica** checkbox.

[“Restoring to a new database in the original availability group” on page 139](#) and [“Restore only on primary replica” on page 144](#) provide details.
13. In the **New database name** box, specify the new database name.

14. In the **Alternate database location** box, specify the path to which to restore the database files, or leave the box blank to restore the files to the original location.

NOTICE

If the path that you specify does not exist, then the restore fails.

15. Select the location to which to restore the log files for the database:
 - To restore the log files to the same location as the database, select **Same as alternate database location** from the **Alternate log location** list.
 - To restore the log files to a different location than the database, select **Different location than database** from the **Alternate log location** list, and then specify the path for the log files in the **Path to alternate log location** box.
16. (Optional) Set other plug-in options as discussed in the following topics:
 - [“Recovery operation options” on page 145](#)
 - [“Authentication options” on page 147](#)
 - [“Point-in-time recovery options” on page 148](#)
17. Disregard the system databases options, which are only necessary when you restore a system database.
18. Click **OK** on the **Restore Command Line Options** dialog box.
19. Click **OK** on the **Restore Options** dialog box.

The **Restore Request** dialog box indicates that the restore started.
20. Click **Close**.

NOTICE

If you chose to perform a tail-log backup and the tail-log backup fails to complete, then the restore cannot take place. Review the log file to determine the cause of the failure. Correct the problem, and then restart the restore. If you clear the **Tail-log backup** checkbox to prevent the tail-log backup from occurring, then the restore includes only the transactions up to the selected backup. You may lose transactions in the tail of the log.

Restoring to a different instance

To restore one or more instances, databases, filegroups, or files to a different SQL Server instance on either the same SQL Server client or a different SQL Server client:

1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 96](#).
2. In Avamar Administrator, click the **Backup & Restore** launcher button.

The **Backup, Restore and Manage** window appears.

3. Find the backup and select the instance, database, filegroup, or file to restore, as discussed in the following topics:

- [“How to find a backup by date” on page 98](#)
- [“How to find a backup for an instance or database” on page 102](#)

4. Select **Actions > Restore Now**.

The **Restore Options** dialog box appears.

Restore Options

Backup Label: Fullsql23
Backup Plug-in: Windows SQL

Restore Destination Client

/Domain/avamar-1.example.com

Restore Plug-in: Windows SQL

Encryption method: High

Restore Destination Choices

Restore to the original location
 Restore to a different SQL Server instance or location
 Restore SQL Server backup as files to the file system

Select the SQL instance on the destination client where the items should be restored.

Items Marked for Restore

Target
Entire contents of backup

5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.

6. Select the encryption method to use for client/server data transfer during the restore.

The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

7. Specify the destination client in the **Restore Destination Client** box:
 - To restore to the original client, leave the default setting of the original client domain and name. (You will select a different instance in the next step.)
 - To restore to a different client, click **Browse** and then browse to the client.
8. Select **Restore to a different SQL Server instance or location**.

Backup Label: Fullsql23
Backup Plug-in: Windows SQL

Restore Destination Client

/Domain/avamar-1.example.com Browse...

Restore Plug-in: Windows SQL

Encryption method: High

Restore Destination Choices

Restore to the original location
 Restore to a different SQL Server instance or location
 Restore SQL Server backup as files to the file system

Select the SQL instance on the destination client where the items should be restored.

Browse...

Items Marked for Restore

Logical Name	Type	Filegroup
(local)/model/modeldev	Rows Data	PRIMARY C:\Program Files\Microsoft SQL Server\90\... \modeldev
(local)/model/modellog	Log	C:\Program Files\Microsoft SQL Server\90\... \modellog
(local)/msdb/MSDBData	Rows Data	PRIMARY C:\Program Files\Microsoft SQL Server\90\... \msdb

Click the "Set Destination" button to enter target destination.

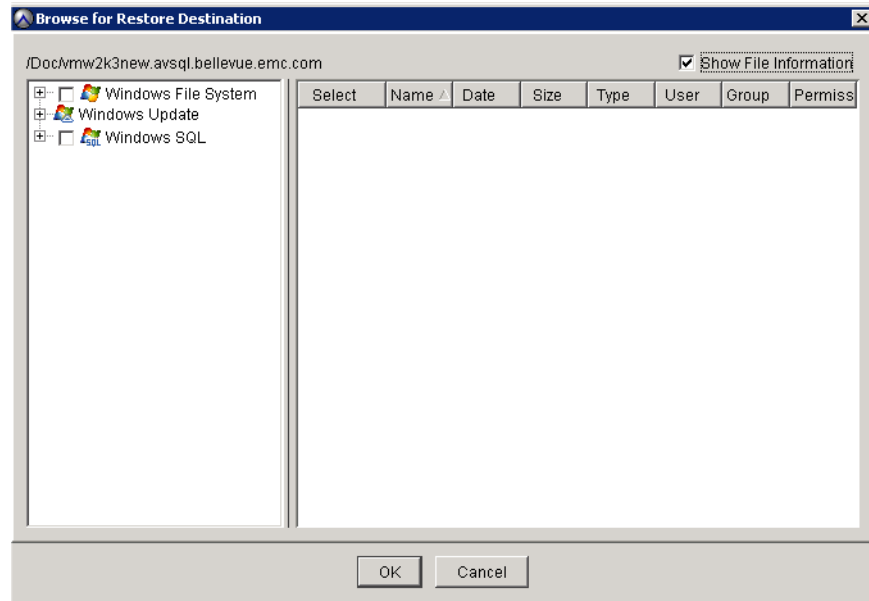
Reset Default Set Destination...

More Options...

OK Cancel Help

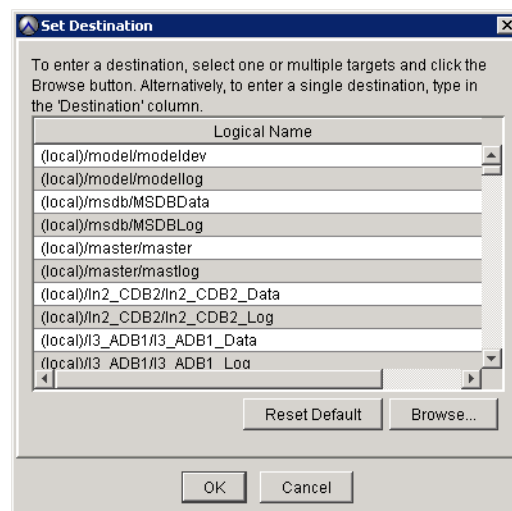
9. Select the instance on the destination client for the restore:
 - a. Next to the **Select the SQL instance on the destination client where the items should be restored** box, click **Browse**.

The **Browse for Restore Destination** dialog box appears.



- b. Select **Windows SQL**.
 - c. In the right pane, select the checkbox next to the instance.
 - d. Click **OK**.
10. Set the destination file path for the database and log files to restore:
 - a. Click **Set Destination**.

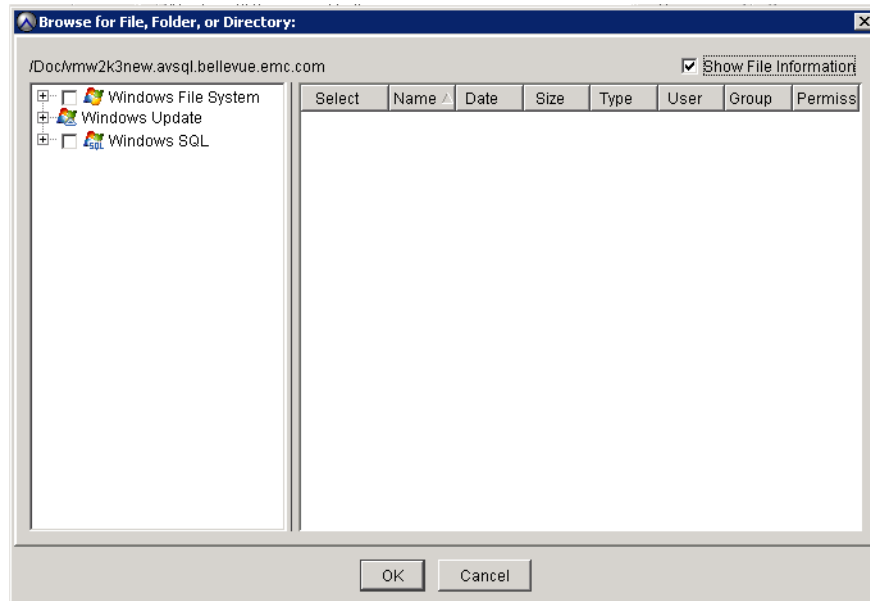
The **Set Destination** dialog box appears.



- b. To specify a path for a single file, select the row in the table. Or, to specify the same path for multiple files in the list, press **Shift** and select the rows.

- c. Click **Browse**.

The **Browse for File, Folder, or Directory** dialog box appears.



- d. Select **Windows File System**.
- e. In the right pane, browse to and select the checkbox for the folder to which to restore the selected files.
- f. Click **OK** to return to the **Set Destination** dialog box.
- g. Repeat [step b](#) through [step f](#) for the remaining rows in the **Set Destination** dialog box.
- h. Click **OK** to return to the **Restore Options** dialog box.
11. Click **More Options**.

The **Restore Command Line Options - Windows SQL** dialog box appears.

12. Select the **Show Advanced Options** checkbox.

Several advanced options appear in red.

The screenshot shows the 'Restore Command Line Options - Windows SQL' dialog box. The 'Show Advanced Options' checkbox is checked. The following options are highlighted in red:

- Use SQL REPLACE option
- Tail-log backup
- Restore only on primary replica
- Recovery operation: RECOVERY
- Standby file location: [text box]
- Enable debugging messages
- System Databases:
 - Restore system databases
 - Manage SQL services automatically during restore
- Redirected Restore:
 - New database name: [text box]
 - Alternate database location: [text box]
 - Alternate log location: Same as alternate database location
 - Path to alternate log location: [text box]
- Authentication:
 - SQL server address: [text box]
 - Authentication method: NT authentication
 - SQL login ID: [text box]
 - SQL password: [text box]
- Point-in-time Recovery:
 - Point-in-time recovery mode: None
 - Point-in-time or mark name string: [text box]
 - Mark recovery point: At mark
 - Mark is after date/time: [text box]

At the bottom, the 'Show Advanced Options' checkbox is checked, and there are 'OK', 'Cancel', and 'Help' buttons.

13. Leave the **Use SQL REPLACE** option clear.
14. Clear the **Tail-log backup** checkbox to disable tail-log backup. Tail-log backup is not supported when you restore to a different SQL Server instance.
15. If you are restoring to an AlwaysOn availability group in a different SQL Server instance, select the **Restore only on primary replica** checkbox to restore the database on only the primary replica.

[“Restoring to a different availability group” on page 140](#) and [“Restore only on primary replica” on page 144](#) provide details.
16. Disregard the **System Databases** options. Do not restore system databases to a different instance.

17. (Optional) To restore the database with a new name, type the new name in the **New database name** box.
18. Disregard the remaining **Redirected Restore** settings (**Alternate database location**, **Alternate log location**, and **Path to alternate log location**). You already specified these settings on the **Set Destination** dialog box.
19. (Optional) Set other plug-in options as discussed in the following topics:
 - [“Recovery operation options” on page 145](#)
 - [“Authentication options” on page 147](#)
 - [“Point-in-time recovery options” on page 148](#)
20. If the target server for the restore has version 6.1 or earlier of the SQL Server plug-in, then you may need to decrease the buffer size that SQL Server uses to read and write backup images. The buffer size increased in version 7.0 of the SQL Server plug-in to improve backup and restore performance.

To decrease the buffer size to the size before version 7.0, set the `--max-transfer-size` attribute to 65536 (64 KB):

 - a. Click **More** on the **Restore Command Line Options** dialog box.
 - b. Type `--max-transfer-size` in the **Enter Attribute** box.
 - c. Type 65536 in the **Enter Attribute Value** box.
 - d. Click **+**.
21. Click **OK** on the **Restore Command Line Options** dialog box.
22. Click **OK** on the **Restore Options** dialog box.

The **Restore Request** dialog box indicates that the restore started.
23. Click **Close**.

Restoring to a file

If the Avamar Plug-in for SQL Server is not installed on the target server, or you want to use the standard SQL Server restore tools for features that the SQL Server plug-in does not provide, then you can restore a SQL Server database from an Avamar backup to operating system files. You can use either the Avamar Plug-in for SQL Server or the Avamar Plug-in for the Windows File System to restore a database to a file. You can then use SQL Server tools to restore the data to SQL Server.

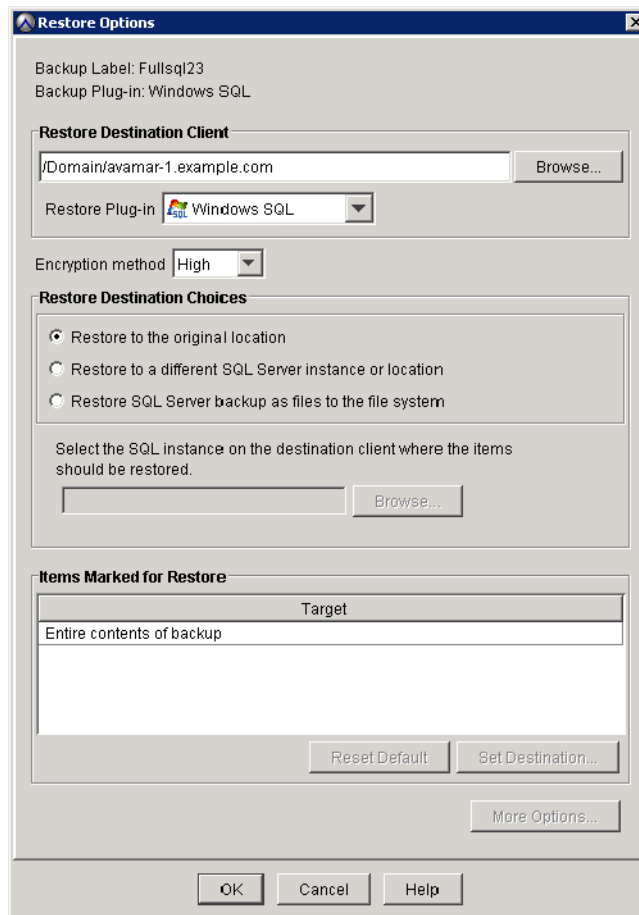
Restoring to a file with the SQL Server plug-in

To restore an instance or database to operating system files by using the Avamar Plug-in for SQL Server:

1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 96](#).
2. In Avamar Administrator, click the **Backup & Restore** launcher button.
The **Backup, Restore and Manage** window appears.
3. Find the backup and select the database to restore, as discussed in the following topics:
 - [“How to find a backup by date” on page 98](#)
 - [“How to find a backup for an instance or database” on page 102](#)

4. Select **Actions > Restore Now**.

The **Restore Options** dialog box appears.



5. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.

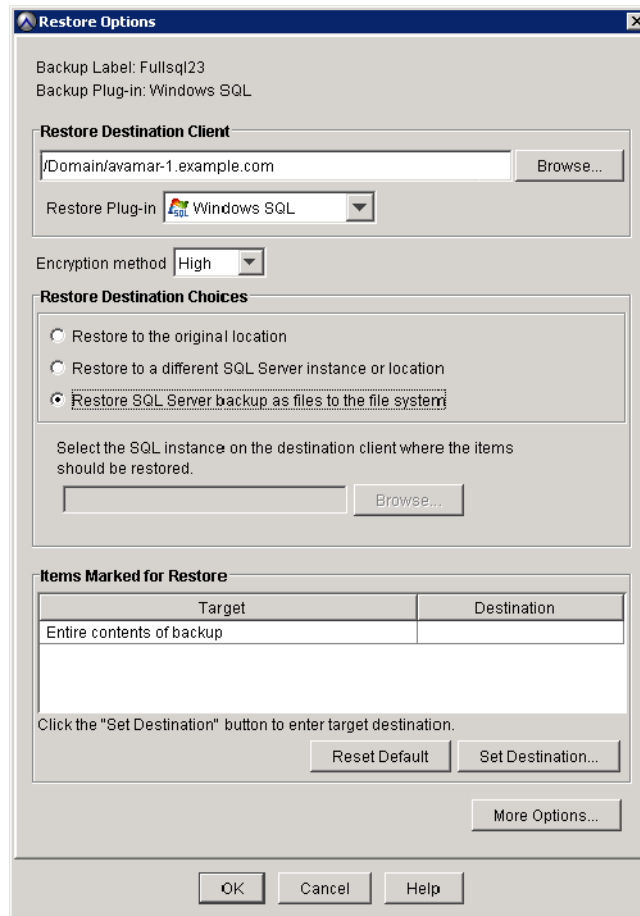
6. Select the encryption method to use for client/server data transfer during the restore.

The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

7. Specify the destination client in the **Restore Destination Client** box:

- To restore to the original client, leave the default setting of the original client domain and name. (You will select a different instance in the next step.)
- To restore to a different client, click **Browse** and then browse to the client.

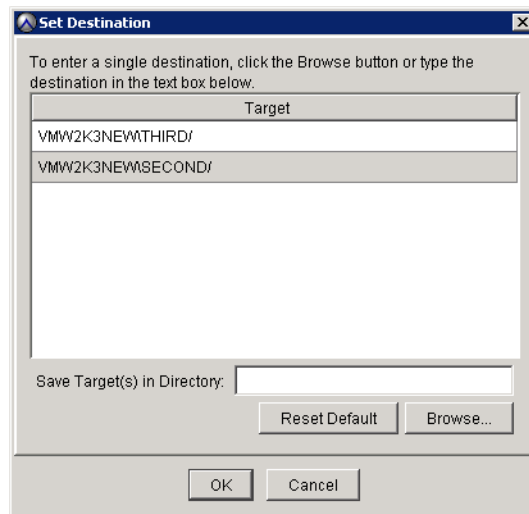
8. Select **Restore SQL Server backup as files to the file system**.



9. Set the destination file path for the database and log files to restore:

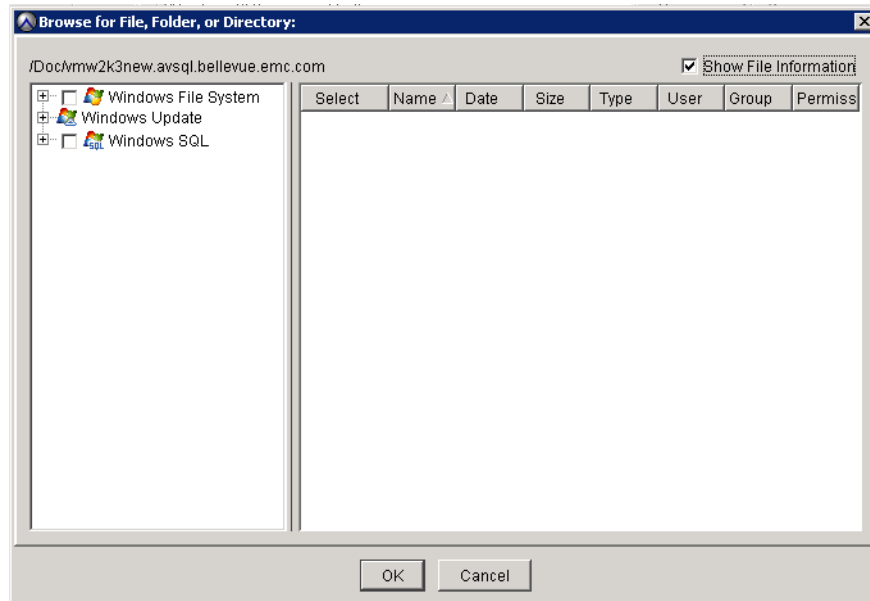
a. Click **Set Destination**.

The **Set Destination** dialog box appears.



- b. Click **Browse**.

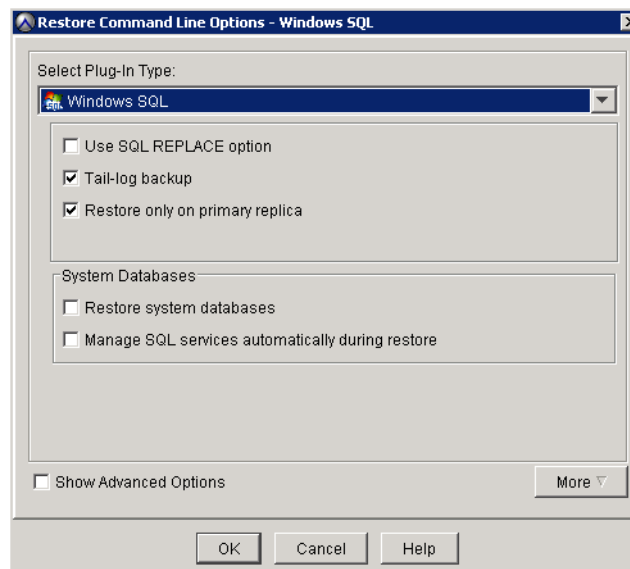
The **Browse for File, Folder, or Directory** dialog box appears.



- c. Select **Windows File System**.
- d. In the right pane, browse to and select the checkbox for the folder to which to restore the files.
- e. Click **OK** to return to the **Set Destination** dialog box.
- f. Click **OK** to return to the **Restore Options** dialog box.
10. If the restore includes system databases, enable the **Restore system databases** option:

- a. Click **More Options**.

The **Restore Command Line Options - Windows SQL** dialog box appears.



- b. Select the **Restore system databases** checkbox, and disregard the remaining options, which do not apply when you restore to a file.
 - c. Click **OK**.
11. Click **OK**.

The **Restore Request** dialog box indicates that the restore started.

12. Click **Close**.

The backup restores as one or more files to the specified destination in the following path:

destination\instance\database\file

where:

- *destination* is the destination for the files that you specified in the **Set Destination** dialog box.
- *instance* is the name of the SQL Server instance from the backup.
- *database* is the name of the database from the backup.
- *file* is the name of the file. A single backup can include multiple files, depending on the number of streams in the backup. The file name for each file is composed of the backup type and the stream number:
 - f-0 indicates a full backup
 - d-*n* indicates a differential backup
 - i-*n* indicates a transaction log (incremental) backup

where *n* is the sequential number of the differential or incremental backup since the preceding full backup.

For example, if there are two files, *f-0.stream0* and *f-0.stream1*, then the backup files are for a full backup that was sent using two streams.

13. Ensure that the SQL backup format files that you restored are accessible to SQL Server. You may need to make the data visible to SQL Server or copy the data.
14. Manually restore the database by using SQL Server tools. [“Restoring a database with SQL Server tools” on page 126](#) provides instructions.

Restoring to a file with the Windows File System plug-in

To restore an instance or database to operating system files using the Avamar Plug-in for the Windows File System:

1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 96](#).
2. In Avamar Administrator, click the **Backup & Restore** launcher button.
The **Backup, Restore and Manage** window appears.
3. Find the backup and select the database to restore:
 - [“How to find a backup by date” on page 98](#)
 - [“How to find a backup for an instance or database” on page 102](#)

4. Select **Actions > Restore Now**.

The **Restore Options** dialog box appears.

5. Select **Windows File System** from the **Restore Plug-in** list.

Backup Label: Fullsql23
Backup Plug-in: Windows SQL

Restore Destination Client

/Domain/avamar-1.example.com Browse...

Restore Plug-in: Windows File System...

Encryption method: High

Restore Destination Choices

Restore everything to its original location
 Restore everything to a different location
 Restore everything to multiple locations

Items Marked for Restore

Target	Destination
VMW2K3NEWATHIRD/	
VMW2K3NEWSECONDI/	

Click the "Set Destination" button to enter target destination.

Reset Default Set Destination...

More Options...

OK Cancel Help

6. Select the encryption method to use for client/server data transfer during the restore.

The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

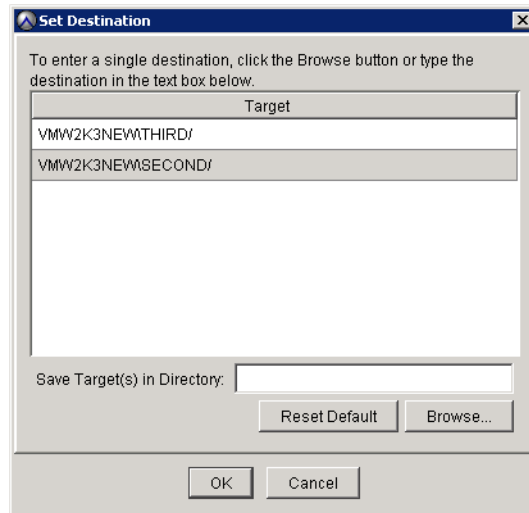
7. Specify the destination client in the **Restore Destination Client** box:

- To restore to the original client, leave the default setting of the original client domain and name. (You will select a different instance in the next step.)
- To restore to a different client, click **Browse** and then browse to the client.

8. Select **Restore everything to a different location**.

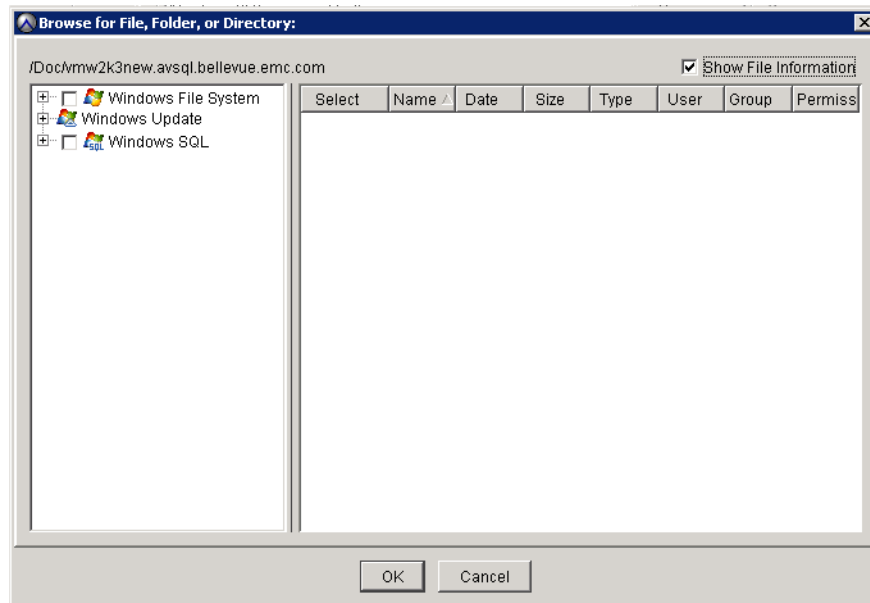
9. Set the destination file path for the database and log files to restore:
 - a. Click **Set Destination**.

The **Set Destination** dialog box appears.



- b. Click **Browse**.

The **Browse for File, Folder, or Directory** dialog box appears.



- c. Select **Windows File System**.
 - d. In the right pane, browse to and select the checkbox for the folder to which to restore the files.
 - e. Click **OK** to return to the **Set Destination** dialog box.
 - f. Click **OK** to return to the **Restore Options** dialog box.
10. (Optional) Click **More Options** and set plug-in options for the restore. The *EMC Avamar for Windows Server User Guide* provides details on the available plug-in options.

11. Click **OK**.

The **Restore Request** dialog box indicates that the restore started.

12. Click **Close**.

The backup restores as one or more files to the specified destination in the following path:

destination\database\file

where:

- *destination* is the destination for the files that you specified in the Set Destination dialog box.
- *database* is the name of the database from the backup.
- *file* is the name of the file. A single backup can include multiple files, depending on the number of streams in the backup. The file name for each file is composed of the backup type and the stream number:
 - f-0 indicates a full backup
 - d-*n* indicates a differential backup
 - i-*n* indicates a transaction log (incremental) backup

where *n* is the sequential number of the differential or incremental backup since the preceding full backup.

For example, if there are two files, *f-0.stream0* and *f-0.stream1*, then the backup files are for a full backup that was sent using two streams.

13. Ensure that the SQL backup format files that you restored are accessible to SQL Server. You may need to make the data visible to SQL Server or copy the data.

14. Manually restore the database by using SQL Server tools. [“Restoring a database with SQL Server tools” on page 126](#) provides instructions.

Restoring a database with SQL Server tools

After you use either the SQL Server plug-in or the Windows file system plug-in to restore a SQL Server backup to a file, you can use SQL Server tools to restore a database to SQL Server. Supported methods include SQL Server Management Studio, a Transact-SQL RESTORE command, and the `sqlcmd` utility.

Restoring a database with SQL Server Management Studio

You can restore a database from a SQL formatted backup file to SQL Server using the user interface in SQL Server Management Studio. The Microsoft website provides full details on how to use SQL Server Management Studio to restore a database backup.

This procedure provides details on using SQL Server Management Studio for SQL Server 2008. The steps for other SQL Server versions may be different.

To restore a database with SQL Server Management Studio using SQL formatted backup files:

1. Restore the database backup to a file. “Restoring to a file with the SQL Server plug-in” on page 119 and “Restoring to a file with the Windows File System plug-in” on page 123 provides instructions.
2. Ensure that the SQL backup format files that you restored are accessible to SQL Server. You may need to make the data visible to SQL Server or copy the data.
3. Restore the full backup (f-0 file) to SQL Server:
 - a. Open the **Restore Database** window:
 - If the database already exists, then right-click the database in the **Object Explorer** and select **Tasks > Restore > Database**.
 - If the database has been lost, then right-click the **Databases** node in the **Object Explorer** and select **Restore Database**.
 - b. On the **General** page of the **Restore Database** window, select **From device**.
 - c. Click the ... button.

The **Specify Backup** dialog box appears.
 - d. Click **Add**.

The **Locate Backup File** dialog box appears.
 - e. Select the folder in which the full backup files are located.
 - f. From the **Files of type** list, select **All files(*)**.
 - g. Select the full backup (f-0) file.
 - h. Click **OK**.
 - i. If there are multiple full backup files from multi-streaming (such as `f-0.stream0`, `f-0.stream1`, `f-0.stream2`, and so on), then repeat [step d](#) through [step h](#) to add each file.
 - j. Click **OK** on the **Specify Backup** dialog box.
 - k. On the **General** page of the **Restore Database** window, select the checkboxes next to the backup files to restore.
 - l. In the left pane, click **Options** to open the **Options** page.
 - m. In the **Restore the database files as** list, select each file and click the ... button to specify the location to which to restore the files.
 - n. For **Recovery state**, select **RESTORE WITH NORECOVERY**.
 - o. Click **OK** to begin the restore.

4. Restore the differential (d-*n*) or transaction log (i-*n*) files in order from the oldest to the most recent:
 - a. In the **Object Explorer**, right-click the database and select **Tasks > Restore > Database**.
 - b. On the **General** page of the **Restore Database** window, select **From device**.
 - c. Click the ... button.

The **Specify Backup** dialog box appears.
 - d. Click **Add**.

The **Locate Backup File** dialog box appears.
 - e. Select the folder in which the differential or transaction log backup files are located.
 - f. From the **Files of type** list, select **All files(*)**.
 - g. Select the differential (d-*n*) or transaction log (i-*n*) backup file, where *n* is the sequential number of the differential or incremental backup since the preceding full backup.
 - h. Click **OK**.
 - i. If there are multiple differential or transaction log backup files from multi-streaming (such as d-3.stream0, d-3.stream1, and d-3.stream2, or i-6.stream0, i-6.stream1, i-6.stream2, and i-6.stream3), then repeat [step d](#) through [step h](#) to add each file.
 - j. Click **OK** on the **Specify Backup** dialog box.
 - k. On the **General** page of the **Restore Database** window, select the checkboxes next to the backup files to restore.
 - l. In the left pane, click **Options** to open the **Options** page.
 - m. In the **Restore the database files as** list, select each file and click the ... button to specify the location to which to restore the files.
 - n. For **Recovery state**, select **RESTORE WITH NORECOVERY** for all except the most recent backup file. When you restore the most recent backup file, select **RESTORE WITH RECOVERY**.
 - o. Click **OK** to begin the restore.
5. If the database is not already listed in SQL Server Management Studio, then refresh the list or connect to the database.

Restoring a database with a Transact-SQL RESTORE command

You can issue a Transact-SQL `RESTORE` command to restore a database from a SQL formatted backup file to SQL Server. The Microsoft website provides full details on how to issue a Transact-SQL command, including details on the available options for the `RESTORE` command.

To restore a database with a Transact-SQL `RESTORE` command:

1. Restore the database backup to a file as described in either [“Restoring to a file with the SQL Server plug-in” on page 119](#) or [“Restoring to a file with the Windows File System plug-in” on page 123](#).
2. Ensure that the SQL backup format files that you restored are accessible to SQL Server. You may need to make the data visible to SQL Server or copy the data.
3. Restore the full backup (f-0 file) to SQL Server using a Transact-SQL command similar to the following command:

```
RESTORE DATABASE dbname
FROM DISK = 'drive:\folder\f-0.stream0'
WITH
  MOVE 'dbname_data' TO 'drive:\folder\dbname.mdf',
  MOVE 'dbname_logfile' TO 'drive:\folder\dbname.ldf',
  NORECOVERY
```

where:

- *dbname* is the name of the database to restore.
 - The `FROM DISK` clause specifies the backup file from which to restore, and *drive:\folder\f-0.stream0* is the path and file name of the full backup file.
 - The `MOVE` clauses specify the path and file name for the restored files:
 - *dbname_data* is the name of the database data file in the backup.
 - *drive:\folder\dbname.mdf* is the path and file name for the restored database file.
 - *dbname_logfile* is the name of the database log file in the backup.
 - *drive:\folder\dbname.ldf* is the path and file name for the restored database log file.
 - The `NORECOVERY` option specifies that the database will remain in the restoring state, which enables you to restore additional backups before setting the database online.
4. Restore all but the most recent differential (d-*n*) or transaction log (i-*n*) backup files, starting with the oldest file, using a Transact-SQL command similar to the command in the previous step. However, replace the *f-0.stream0* file name in the `FROM DISK` clause with the file name for the differential or transaction log backup file.

5. Restore the most recent differential or transaction log backup file using a Transact-SQL command similar to the following command:

```
RESTORE DATABASE dbname
FROM DISK = 'drive:\folder\i-7.stream0'
WITH
  MOVE 'dbname_data' TO 'drive:\folder\dbname.mdf',
  MOVE 'dbname_logfile' TO 'drive:\folder\dbname.ldf',
RECOVERY
```

where:

- *dbname* is the name of the database to restore.
 - The **FROM DISK** clause specifies the backup file from which to restore, and *drive:\folder\i-7.stream0* is the path and file name of the most recent transaction log backup file.
 - The **MOVE** clauses specify the path and file name for the restored files:
 - *dbname_data* is the name of the database data file in the backup.
 - *drive:\folder\dbname.mdf* is the path and file name for the restored database file.
 - *dbname_logfile* is the name of the database log file in the backup.
 - *drive:\folder\dbname.ldf* is the path and file name for the restored database log file.
 - The **RECOVERY** option sets the database online after the restore.
6. If the database is not already listed in SQL Server Management Studio, then refresh the list or connect to the database.

Restoring a database with the sqlcmd utility

The `sqlcmd` utility is a command line utility for entering Transact-SQL statements and scripts. You can restore a database from a backup file by specifying the Transact-SQL `RESTORE` statements with the `sqlcmd` utility. The Microsoft website provides full details on using the `sqlcmd` utility.

NOTICE

You can use the Microsoft SQL Server `osql` utility, but Microsoft recommends the `sqlcmd` utility.

The following example shows how to use the `sqlcmd` utility to restore a single database from a single full backup file:

```
sqlcmd -S server -E
1> restore database dbname
2> from disk = 'drive:\folder\f-0.stream0'
3> with recovery;
4> go
```

where:

- ◆ *server* is the server that is running SQL Server, and optionally, the instance, to which to restore the backup.
- ◆ *dbname* is the database to restore.
- ◆ *drive:\folder\f-0.stream0* is the path and file name of the backup file from which to restore the database.

The following example shows how to use a `restore filelistonly` statement in the `sqlcmd` utility to determine the number and names of the files in the database to restore:

```
sqlcmd -S server -E
1> restore filelistonly
2> from disk = 'drive:\folder\f-0.stream0'
3> go
1> restore database dbname
2> from disk = 'drive:\folder\f-0.stream0'
3> with norecovery,
4> move 'dbname_data' to 'drive:\dbname.mdf',
5> move 'dbname_log' to 'drive:\dbname.ldf'
6> go
```

To restore a database to a point in time, use STOPAT syntax, as shown in the following example:

```
RESTORE DATABASE dbname
FROM disk= 'drive:\folder\f-0.stream0'
WITH NORECOVERY, STOPAT = 'Apr 25, 2013 12:00 AM'
go
RESTORE LOG dbname
FROM disk= 'drive:\folder\l-1.stream0'
WITH RECOVERY, STOPAT = 'Apr 25, 2013 12:00 AM'
go
```

where:

- ◆ *dbname* is the database to restore.
- ◆ *drive:\folder\f-0.stream0* is the path and file name of the full backup file from which to restore the database.
- ◆ *drive:\folder\l-1.stream0* is the path and file name of the transaction log backup file from which to restore.

Restoring system databases

It is rare that you need to restore only system databases. One possible scenario for restoring only system databases is if damage occurs to one or more system databases.

When you restore system databases, the Avamar Plug-in for SQL Server can automatically restore the databases in the correct order and manage SQL Server services. If necessary, however, you can restore individual system databases and manually manage the services.

Restoring system databases automatically

When you restore multiple system databases, the Avamar Plug-in for SQL Server automatically restores the databases in the correct order—master, msdb, and model. The SQL Server plug-in can also automatically manage the stop and restart of the necessary SQL Server services during the restore.

To restore a system database:

1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 96](#).
2. Close all instances of SQL Server Management Studio, and disable any other possible connections to the system databases.

If there are other connections to the system databases, then Avamar may not be able to restore the master database.

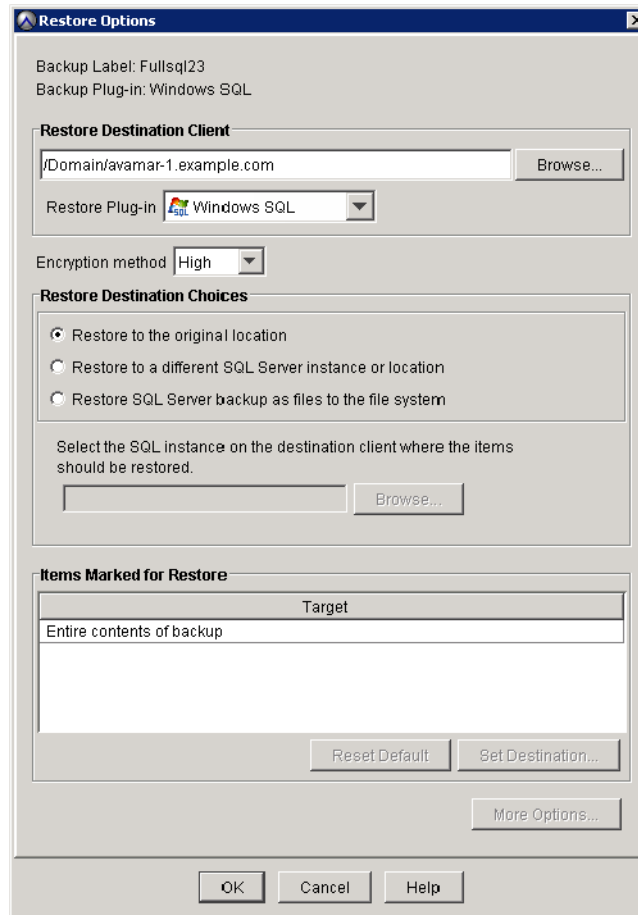
3. In Avamar Administrator, click the **Backup & Restore** launcher button.

The **Backup, Restore and Manage** window appears.

4. Find the backup and select the system databases to restore, as discussed in the following topics:
 - [“How to find a backup by date” on page 98](#)
 - [“How to find a backup for an instance or database” on page 102](#)

5. Select **Actions > Restore Now**.

The **Restore Options** dialog box appears.



6. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.

7. Select the encryption method to use for client/server data transfer during the restore.

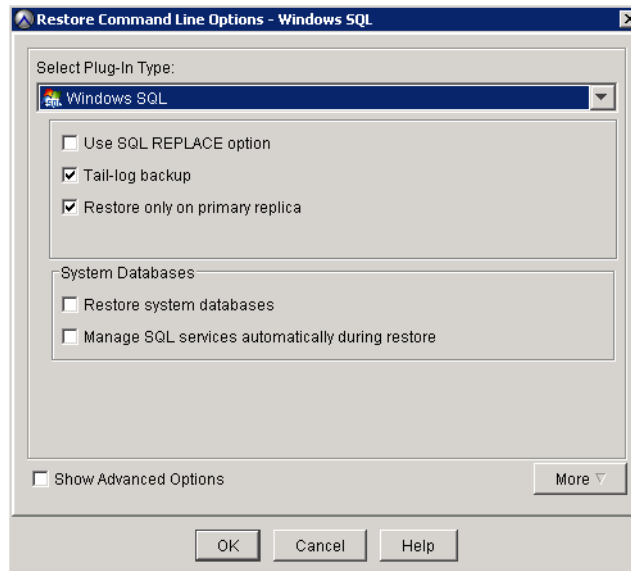
The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

8. Leave the default selection of **Restore to the original location**.

If you plan to restore the system databases to a file, follow the steps in [“Restoring to a file” on page 119](#) instead of the steps in this procedure.

9. Click **More Options**.

The **Restore Command Line Options - Windows SQL** dialog box appears.

10. Select the **Use SQL REPLACE option** checkbox.11. Clear the **Tail-log backup** checkbox.12. Select the **Restore system databases** checkbox.13. Select the **Manage SQL services automatically during restore** checkbox.14. (Optional) Select the **Show Advanced Options** checkbox, and set other plug-in options as discussed in the following topics:

- [“Recovery operation options” on page 145](#)
- [“Authentication options” on page 147](#)

15. Disregard the redirected restore options, which are only necessary when you are restoring to a different location. In addition, disregard the point-in-time recovery options, which are only supported for databases that use the full recovery model.

16. Click **OK** on the **Restore Command Line Options** dialog box.17. Click **OK** on the **Restore Options** dialog box.

The **Restore Request** dialog box indicates that the restore started.

18. Click **Close**.

Restoring system databases manually

When you restore system databases manually, you must manage the services and restore the databases in the correct order.

To restore the system databases manually:

1. Shut down the SQL Server instance, and ensure that dependent services stop, such as the SQL Server Agent service and the Analysis Service.
2. Close all instances of SQL Server Management Studio, and disable any other possible connections to the system databases.

If there are other connections to the system databases, then Avamar may not be able to restore the master database.

3. Start the SQL Server instance in single-user mode by running the `sqlservr.exe` application with the `-m` and `-c` options:

- To start the default instance in single-user mode, open a command prompt and type the following command:

```
cd \MSSQLPATH\Binn
sqlservr.exe -m -c
```

- To start a named instance in single-user mode, open a command prompt and type the following command:

```
cd \MSSQLPATH\Binn
sqlservr.exe instance -m -c
```

where `\MSSQLPATH\Binn` is the path to the `Binn` folder for the instance, and `instance` is the name of the instance to start.

4. Wait for the **Recovery Complete** message to appear on the console.

If you log in to the SQL Server computer as a local or domain administrator with the SQL services running under the Local System account, then the `sqlservr.exe` command may fail to correctly start SQL services in single-user mode. If that occurs, complete [step a](#) through [step e](#) instead of running `sqlservr.exe` from the command line. Otherwise, skip [step a](#) through [step e](#) and go to [step 5](#).

Complete the following steps if SQL services do not start in single-user mode correctly:

- a. Shut down the SQL service. If SQL Server is installed on a standalone server, then use the Windows Services console. If SQL Server is installed in a cluster, use Cluster Manager.
- b. Right-click the SQL service in the Windows **Services** console, and then click **Properties**.
- c. In the **Start parameters** box, type `-m -c`.
- d. Click **Start** to start the service.
- e. Click **OK** to close the **Properties** dialog box.

5. Restore the master database to the original location:
 - a. Ensure that the environment meets the guidelines in [“Restore requirements” on page 96](#).
 - b. In Avamar Administrator, click the **Backup & Restore** launcher button.
The **Backup, Restore and Manage** window appears.
 - c. Find the backup and select the master database to restore, as discussed in the following topics:
 - [“How to find a backup by date” on page 98](#)
 - [“How to find a backup for an instance or database” on page 102](#)
 - d. Select **Actions > Restore Now**.
The **Restore Options** dialog box appears.
 - e. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
 - f. Select the encryption method to use for client/server data transfer during the restore.

The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.
 - g. Leave the default selection of **Restore to the original location**.
 - h. Click **More Options**.
The **Restore Command Line Options - Windows SQL** dialog box appears.
 - i. Select the checkbox next to the following options:
 - **Use SQL REPLACE option**
 - **Restore system databases**
 - j. Clear the **Tail-log backup** checkbox.
 - k. (Optional) Select **Show Advanced Options**, and set the authentication options as discussed in [“Authentication options” on page 147](#).
 - l. Disregard the remaining restore options, which do not apply when you restore the master database.
 - m. Click **OK** on the **Restore Command Line Options** dialog box.
 - n. Click **OK** on the **Restore Options** dialog box.
The **Restore Request** dialog box indicates that the restore started.
 - o. Click **Close**.

After you restore the master database, the SQL Server service is stopped automatically.

6. Restart the SQL Server service:
 - To start the default instance of SQL Server, open a command prompt and type the following command:


```
net start MSSQLServer
```
 - To start a named instance of SQL Server, open a command prompt and type the following command:


```
net start MSSQL$instance
```

 where *instance* is the name of the instance.
7. Restore the msdb and model databases:
 - a. Ensure that the environment meets the guidelines in [“Restore requirements” on page 96](#).
 - b. In Avamar Administrator, click the **Backup & Restore** launcher button.

The **Backup, Restore and Manage** window appears.
 - c. Find the backup and select the msdb and model databases to restore, as discussed in the following topics:
 - [“How to find a backup by date” on page 98](#)
 - [“How to find a backup for an instance or database” on page 102](#)
 - d. Select **Actions > Restore Now**.

The **Restore Options** dialog box appears.
 - e. Ensure that **Windows SQL** appears in the **Restore Plug-in** list.
 - f. Select the encryption method to use for client/server data transfer during the restore.

The encryption technology and bit strength for a client/server connection depend on several factors, including the client operating system and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.
 - g. Leave the default selection of **Restore to the original location**.
 - h. Click **More Options**.

The **Restore Command Line Options - Windows SQL** dialog box appears.
 - i. Select the checkbox next to the following options:
 - **Use SQL REPLACE option**
 - **Restore system databases**
 - j. Clear the **Tail-log backup** checkbox.
 - k. (Optional) Select **Show Advanced Options**, and set the authentication options as discussed in [“Authentication options” on page 147](#).
 - l. Disregard the remaining restore options, which do not apply when you restore the master database.
 - m. Click **OK** on the **Restore Command Line Options** dialog box.

- n. Click **OK** on the **Restore Options** dialog box.
The **Restore Request** dialog box indicates that the restore started.
 - o. Click **Close**.
8. If required, restart the SQL Server Agent service:
 - To start the default instance of SQL Server Agent, open a command prompt and type the following command:
net start SQLSERVERAGENT
 - To start a named instance of SQL Server Agent, open a command prompt and type the following command:
net start SQLAGENT\$instance
where *instance* is the name of the instance.

Restoring to an AlwaysOn availability group

When you back up databases in an AlwaysOn availability group, you can restore the databases to the following locations:

- ◆ The original location in the original availability group
- ◆ A new database in the original availability group
- ◆ A different availability group
- ◆ A SQL Server instance that does not use availability groups
- ◆ Operating system files

Restoring to the original availability group

When you restore to the original location in an AlwaysOn availability group, the restore process can automatically restore the databases on both the primary replica and the secondary replicas. You can also restore databases only on the primary replica.

Restoring automatically to both primary and secondary replicas

To automatically restore the databases to both the primary replica and the secondary replicas, perform the steps in [“Restoring to the original location” on page 106](#). During the restore, set the following values:

- ◆ Select the client for the availability group listener when you search for a backup to restore.
- ◆ Set the plug-in options with the following values:
 - Clear the **Restore only on primary replica** checkbox.
 - Leave the **System Databases** options clear. SQL Server does not support system databases in an availability group.
 - (Optional) Perform a tail-log backup or use SQL `REPLACE` to force a restore even if the database already exists.
 - Disregard the **Redirected Restore** options.

- (Optional) Specify authentication information.
- (Optional) Perform a point-in-time recovery.

Restoring to only the primary replica

To restore the databases to only the primary replica:

1. Perform the steps in [“Restoring to the original location” on page 106](#). During the restore, set the following values:
 - Select the client for the availability group listener when you search for a backup to restore.
 - Set the plug-in options with the following values:
 - Leave the **Restore only on primary replica** checkbox selected.
 - Leave the **System Databases** options clear. SQL Server does not support system databases in an availability group.
 - (Optional) Perform a tail-log backup or use SQL REPLACE to force a restore even if the database already exists.
 - Disregard the **Redirected Restore** options.
 - (Optional) Specify authentication information.
 - (Optional) Perform a point-in-time recovery.

After you restore a database on only the primary replica, the corresponding database on the secondary replicas is in a restoring state.

2. (Optional) Restore the databases on the secondary replicas by manually preparing and restoring the databases, and then joining them to the availability group on the secondary replicas.

Restoring to a new database in the original availability group

You can restore a database in an AlwaysOn availability group to its original instance but with a new name, creating a new database in the instance.

To automatically restore the databases to both the primary replica and the secondary replicas, perform the steps in [“Restoring to a new database in the original instance” on page 109](#). During the restore, set the following values:

- ◆ Select the client for the availability group listener when you search for a backup to restore.
- ◆ Set the plug-in options with the following values:
 - Leave the **Use SQL REPLACE option** checkbox clear.
 - (Optional) Perform a tail-log backup.
 - Clear the **Restore only on primary replica** checkbox.
 - (Optional) Specify recovery operation options.
 - Leave the **System Databases** options clear. SQL Server does not support system databases in an availability group.

- Specify a name for the new database in the **New database name** box.
- (Optional) Specify a new path for the database files in the **Alternate database location** box.
- (Optional) Select the location for the restored database log files.
- (Optional) Specify authentication information.
- (Optional) Perform a point-in-time recovery.

Restoring to a different availability group

To restore the databases to an availability group in a different cluster:

1. Perform the steps in [“Restoring to a different instance” on page 113](#). During the restore, set the following values:
 - Select the client for the original availability group listener when you search for a backup to restore.
 - Set the plug-in options with the following values:
 - Leave the **Use SQL REPLACE option** checkbox clear.
 - Clear the **Tail-log backup** checkbox. This restore scenario does not support tail-log backup.
 - Leave the **Restore only on primary replica** checkbox selected.
 - (Optional) Specify recovery operation options.
 - Leave the **System Databases** options clear. SQL Server does not support system databases in an availability group.
 - Disregard the **Redirected Restore** options. You already specified these settings on the **Set Destination** dialog box.
 - (Optional) Specify authentication information.
 - (Optional) Perform a point-in-time recovery.
2. After the restore completes, join the new database to the availability group.

When you add the database to the availability group, the process automatically creates and synchronizes the databases on the secondary replicas.

Restoring to a SQL Server instance without availability groups

When you back up databases in an availability group, you can restore the databases to a different SQL Server instance that does not use availability groups, either on the same server or on a different server. The steps are the same whether you restore from a backup of databases on a stand-alone server, on shared storage in a failover cluster, or in an availability group. [“Restoring to a different instance” on page 113](#) provides details.

Restoring to operating system files

When you back up databases in an availability group, you can restore the databases to operating system files. The steps are the same whether you restore from a backup of databases on a stand-alone server, on shared storage in a failover cluster, or in an availability group. [“Restoring to a file” on page 119](#) provides details.

Restoring a database with an intact log file

If a database becomes corrupt or is otherwise lost but an intact database log file is available, you can restore the database and use the log file to recover transactions after the most recent backup.

To restore a database with an intact log file:

1. Perform a transaction log backup of the intact database log by issuing the following Transact-SQL command:

```
BACKUP LOG dbname TO DISK = 'drive:\folder\filename' WITH NO_TRUNCATE
```

where *dbname* is the name of the database and *drive:\folder\file* is the path to the folder and file name to which to save the backup.

2. In Avamar Administrator, find the most recent backup of the database by using the instructions in [“How to find a backup for an instance or database” on page 102](#).
3. In Avamar Administrator, restore the backup to its original location using the instructions in [“Restoring to the original location” on page 106](#).

On the **Restore Command Line Options** dialog box during the restore, select the **Show Advanced Options** checkbox, and then configure the settings as follows:

- Select the **Use SQL REPLACE option** checkbox.
 - Clear the **Tail-log backup** checkbox.
 - Select **NORECOVERY** from the **Recovery operation** list.
 - Disregard the remaining options, which do not apply to this restore.
4. Restore the transaction log backup of the intact database log by issuing the following Transact-SQL command:


```
RESTORE LOG dbname FROM DISK = 'drive:\folder\filename' WITH RECOVERY
```

where *dbname* is the name of the database and *drive:\folder\file* is the path and file name for the transaction log backup file.
 5. If the database does not already appear in SQL Server Management Studio, then refresh the list or connect to the database.

Setting restore options

You set plug-in options during the restore using the **Restore Command Line Options** dialog box. To view all plug-in options, including advanced options, select the **Show Advanced Options** checkbox. The advanced options appear in red.

Figure 10 Restore Command Line Options - Windows SQL dialog box

General restore options

General options on the **Restore Command Line Options** dialog box include the **Use SQL REPLACE option** checkbox, the **Tail-log backup** checkbox, and the **Enable debugging messages** checkbox.

Use SQL REPLACE option

When you select the **Use SQL REPLACE option** checkbox on the **Restore Command Line Options** dialog box, an SQL `WITH REPLACE` clause statement is added to the restore Transact-SQL command. The statement specifies that SQL Server will create any necessary database and related files even if another database or file already exists with the same name.

NOTICE

This option overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The Microsoft Transact-SQL documentation [provides more information in the RESTORE command section](#).

Select the checkbox only when you are restoring an instance, database, filegroup, or file to its original location and you need to force the overwrite of the original data. You may need to force the overwrite if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log:

```
One or more devices or files already exist. Reissue the statement using
the WITH REPLACE option to overwrite these files and devices.
```

Use this option also when you restore system databases.

Tail-log backup

When you select the **Tail-log backup** checkbox on the **Restore Command Line Options** dialog box, Avamar backs up the tail of the transaction log during the restore to capture the log records that are not in a backup. The restore process then uses the tail-log backup after the database restore to recover the transactions that were not in the backup.

To perform a tail-log backup, the database must be online and using either the full or bulk-logged recovery model. As a result, you cannot perform a tail-log backup of system databases such as the master and msdb databases because those databases use the simple recovery model.

When you are restoring a user-defined filegroup or secondary data file and you perform a tail-log backup, you must select the most recent backup as the backup from which to restore. Otherwise, the restore fails and an error message is written to the log file.

You can perform a tail-log backup when you are restoring an instance, database, filegroup, or file to its original location *without* the SQL `WITH REPLACE` option. In other words, if you select the **Use SQL REPLACE option** checkbox when you are restoring to the original location, then leave the **Tail-log backup** checkbox clear.

You can also perform a tail-log backup when you are restoring a database to the original instance but with a new database name.

If you are performing a point-in-time restore and the point in time to which you are restoring is after the most recent transaction log backup, then you *must* perform a tail-log backup.

A tail-log backup is also required if you are restoring a file from a user-defined filegroup to its original location.

Do *not* perform a tail-log backup if you are performing a redirected restore to a different SQL Server instance.

NOTICE

If the tail-log backup fails to complete, then the restore cannot take place. Review the log file to determine the cause of the failure. Correct the problem, and then restart the restore. If you clear the **Tail-log backup** checkbox to prevent the tail-log backup, then the restore includes only the transactions up to the selected backup. You may lose any transactions in the tail of the log.

Restore only on primary replica

The **Restore only on primary replica** checkbox controls whether databases restore to only the primary replica or to the primary replica and all secondary replicas in an AlwaysOn availability group.

When you clear the checkbox, the restore process automatically restores the databases to the original location on both the primary replica and the secondary replicas. Clear the checkbox only when you restore to the original location in the original availability group and you want to restore to all replicas.

When you select the checkbox and restore the databases on only the primary replica, the corresponding database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, you can manually prepare and restore the databases, and then join them to the availability group on the secondary replica.

You can also set the database on a secondary replica online without rejoining it to the availability group by restoring the database with the `RECOVERY` recovery operation. The SQL Server 2012 documentation available on the Microsoft website provides details.

Select the checkbox when you restore to a different location in the original availability group or when you restore to a different availability group. You can also select the checkbox if you want to restore to only the primary replica in the original availability group.

[“Restoring to an AlwaysOn availability group” on page 138](#) provides additional details on restoring in an availability group.

Enable debugging messages

When you select the **Enable debugging messages** checkbox on the **Restore Command Line Options** dialog box, maximum information is written to log files during the restore. If you select the checkbox, then the restore process generates very large log files. Use this option only for debugging purposes.

Recovery operation options

The **Recovery operation** list and **Standby file location** box on the **Restore Command Line Options** dialog box enable you to control the recovery operation that occurs after the restore.

Table 12 Recovery operation options

Recovery operation	Description
RECOVERY	The database is fully recovered and online after the restore. This is the default setting.
NORECOVERY	The database remains in a restoring state after the restore. This enables you to perform additional manual restore tasks, such as applying additional SQL transaction log files.
STANDBY	<p>The database is in standby (read-only) mode after the restore. This enables you to bring up a database for read-only access between transaction log restores, and can be used with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores.</p> <p>This option also creates a file with recovery changes. You can use the file to revert the recovery changes, if required. The size of the file depends on the volume of undo actions from uncommitted transactions. Specify the path to this file in the Standby file location box. Use the following format:</p> <p><i>drive:\folder\subfolder</i></p> <p>where <i>drive</i> is the drive letter on the destination client, and <i>folder\subfolder</i> is the path on the drive letter in which to create the file. If a file with recovery changes already exists in the specified location, then SQL Server overwrites it. If you do not specify a path, then Avamar creates the file in the <i>C:\Program Files\avs\var</i> folder, where <i>C:\Program Files\avs</i> is the Avamar installation folder.</p> <p>Do not use STANDBY when a database upgrade is necessary. You may need to perform a database upgrade when restoring backup sets from an earlier version of SQL Server.</p>

You can specify the recovery operation in the following restore scenarios:

- ◆ You are restoring an instance, database, filegroup, or file to its original location.
- ◆ You are restoring a database to the original instance but with a new database name.
- ◆ You are restoring an instance, database, filegroup, or file to a different instance on either the original server or a different server.
- ◆ You are restoring one or more system databases. Keep in mind the following points, however:
 - If you are restoring the master or model database, then you must select the **RECOVERY** option. Do not use either the **NORECOVERY** or **STANDBY** options.
 - If you are restoring the msdb database, then you can select any of the recovery operation options. However, if you select **NORECOVERY** or **STANDBY**, then all databases become inaccessible until the restore of the msdb database is complete.
 - If you are restoring the system databases, then you must select the **RECOVERY** option. Do not use either the **NORECOVERY** or **STANDBY** options.

You do not need to specify the recovery operation when you are restoring to an operating system file.

If you specify either the `NORECOVERY` or `STANDBY` recovery operation, then you can bring the database back online after any manual operations by using one of the following methods:

- ◆ Restore the database again using the Avamar Plug-in for SQL Server, but select `RECOVERY` from the **Recovery operation** list.
- ◆ Issue a `RESTORE DATABASE dbname WITH RECOVERY` command using SQL Server Management Studio, where `dbname` is the name of the database to restore.

System database restore options

The **Restore system databases** and **Manage SQL services automatically during restore** checkboxes on the **Restore Command Line Options** dialog box enable you to correctly restore system databases, such as the master, msdb, and model databases.

When you restore an entire instance, or if you specifically select system databases for restore, then select the **Restore system databases** checkbox to ensure that the system databases restore. If you leave the checkbox clear, then the system databases do not restore.

The **Manage SQL services automatically during restore** option automatically stops and restarts SQL services during the restore:

- ◆ When you restore the master database, this option automatically stops the SQL Server instance, including dependent services such as the SQL Server agent service and the Analysis Service. The option also restarts the instance in single-user mode before the restore. After the restore, the instance automatically restarts.
- ◆ When you restore the msdb database, this option automatically stops the SQL Server agent service, and then restarts it when the restore is complete.

When you select both system and user databases for restore, the system databases restore first. You must select the **Manage SQL services automatically during restore** checkbox to ensure that all system databases restore in the proper order and with the necessary service stops and restarts.

Redirected restore options

The redirected restore options on the **Restore Command Line Options** dialog box enable you to control the database name and file locations when you restore a database to the original instance but with a new name.

NOTICE

You also can specify the file locations by clicking **Set Destination** on the **Restore Options** dialog box. If you specify the file locations in the **Restore Options** dialog box, then you do not need to specify them on the **Restore Command Line Options** dialog box.

You can also specify a new database name when you are restoring a database to a different instance but with a new name.

To specify redirected restore options:

1. In the **Restore Command Line Options** dialog box, select the **Show Advanced Options** checkbox.
Several advanced options, including the point-in-time recovery options, appear in red.
2. (Optional) If you are restoring a single database with a new name, specify the new name in the **New Database** name box.
3. To restore the database files to a different path than the original path, type the full path of the new database file (*.mdf) location on the client in the **Alternate database location** box. An example path is `C:\temp`.
4. Select the path to which to restore the database log file (*.ldf):
 - To restore the database log file to the same location as the database files (as specified in the **Alternate database location** box), select **Same as alternate database location** from the **Alternate log location** list.
 - To restore the database log file to a different location than the database files, select **Different location** from the **Alternate log location** list, and then type the path to the location on the client in the **Path to alternate log location** box. An example path is `C:\temp\logs`.

Authentication options

Authentication options enable you to specify whether Avamar uses Windows (NT) authentication or SQL Server authentication to connect to SQL Server when you restore an instance, database, filegroup, or file to either its original location or to a different location.

If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account. [“Security requirements” on page 40](#) provides more information.

You do not need to specify authentication options when you are restoring to an operating system file because it is not necessary to connect to SQL Server during this type of restore.

To specify authentication options:

1. In the **Restore Command Line Options** dialog box, select the **Show Advanced Options** checkbox.
Several advanced options, including the authentication options, appear in red.
2. In the **SQL server address** box, specify the hostname or IP address of the computer that is running SQL Server.
3. From the **Authentication method** list, select whether to use **NT authentication** or **SQL Server authentication**.
4. If you select SQL Server authentication, specify the login ID and password for the SQL Server account in the **SQL login ID** and **SQL password** boxes, respectively.

Point-in-time recovery options

If you are restoring a database that uses the full recovery model to either its original location or to a different location, then you can restore to either a specific date and time or to a named mark in the transaction log.

You cannot perform a point-in-time restore of system databases such as the master and msdb databases because those databases use the simple recovery model.

To restore to a specific point in time, you must provide the transaction date and time or named mark to which to recover from the SQL Server transaction log. The SQL Server documentation on the Microsoft website provides details on how to access transaction log information.

The point in time to which you are restoring must be after the finish time for the most recent full backup. In addition, if the point in time is before the start time of the most recent transaction log (incremental) backup, then a tail-log backup is not required. However, a tail-log backup *is* required if the point in time is after the most recent transaction log backup.

When you specify the point in time for restore, do not specify the start time of the selected transaction log backup if it is not the last backup in the backup sequence. Otherwise, the restore fails and a tail-log backup does not occur even if you select the Tail-log backup option.

To perform a point-in-time recovery:

1. In the **Restore Command Line Options** dialog box, select the **Show Advanced Options** checkbox.

Several advanced options, including the point-in-time recovery options, appear in red.
2. From the **Point-in-time recovery mode** list, select whether to recover to a point in time or a named mark:
 - To recover to a point in time, select **Point-in-time**.
 - To recover to a named mark, select **Mark name**.
3. In the **Point-in-time or mark name string** box, specify either the point in time or the named mark to which to recover:
 - To recover to a specific point in time, specify the date and time in yyyy-mm-ddThh:mm:ss format. For example, 2013-10-15T14:15:45 is October 15, 2013 at 2:15:45 p.m.
 - To recover to a named mark, specify the mark.
4. If you specified a mark, select whether to include the mark in the recovery:
 - To specify that the log record immediately before the mark is the recovery point, select **Before mark** from the **Mark recovery point** list. In other words, the recovery rolls forward to the mark and excludes the marked transaction.
 - To specify that the marked transaction is the recovery point, select **At mark** from the **Mark recovery point** list. In other words, the recovery rolls forward to the mark and includes the marked transaction.

- If you specified a mark and named marks are not unique in the transaction log, then use the **Mark is after date/time** box to locate the mark to which to recover. The recovery process stops at the first mark with the specified name, exactly at or after the specified date and time. Specify the date and time in yyyy-mm-ddThh:mm:ss format.

Monitoring restores

- In Avamar Administrator, click the **Activity** tab.

The **Activity** window appears.

- Select the **Activity Monitor** tab.

A list of all activities appears.

Activity retained for 72 hours up to a maximum of 5000 rows.
Waiting-Queued: 0 Waiting-Client: 0 Active: 0 Finished: 202

Status	Error Code	Start Time (PST)	Elapsed	End Time (PST)	Type	Server	Progress Bytes	New Bytes
Completed		2012-12-10 19:27	00h:00m:29s	2012-12-10 19:28	On-Demand Backup	Avamar	4.7 MB	2.4%
Completed		2012-12-12 10:07	00h:00m:58s	2012-12-12 10:08	Restore	Avamar	5.6 MB	11.4%
Completed		2012-12-11 18:51	00h:00m:21s	2012-12-11 18:52	On-Demand Backup	Avamar	18.8 MB	3%
Failed	10007	2012-12-12 15:25	00h:00m:08s	2012-12-12 15:25	On-Demand Backup	Avamar	0	0%
Completed		2012-12-11 23:53	00h:00m:26s	2012-12-11 23:53	On-Demand Backup	Avamar	< 1MB	4.4%
Completed		2012-12-12 08:51	00h:00m:12s	2012-12-12 08:51	Restore	Avamar	4.7 MB	4.2%
Completed		2012-12-12 11:37	00h:00m:21s	2012-12-12 11:38	On-Demand Backup	Avamar	4.9 MB	4%
Completed		2012-12-10 13:58	00h:00m:28s	2012-12-10 13:58	On-Demand Backup	Avamar	4.7 MB	1.4%
Completed		2012-12-12 14:52	00h:00m:47s	2012-12-12 14:53	Restore	Avamar	4.7 MB	9.8%
Completed		2012-12-10 13:22	00h:00m:28s	2012-12-10 13:23	Restore	Avamar	4.7 MB	5.1%
Completed w/Exception(s)	10020	2012-12-10 16:23	00h:00m:27s	2012-12-10 16:24	On-Demand Backup	Avamar	4.7 MB	1.1%
Failed	10007	2012-12-12 19:50	00h:00m:08s	2012-12-12 19:50	On-Demand Backup	Avamar	0	0%
Completed		2012-12-11 23:32	00h:00m:19s	2012-12-11 23:33	On-Demand Backup	Avamar	4.7 MB	1%
Completed		2012-12-12 13:25	00h:00m:31s	2012-12-12 13:25	On-Demand Backup	Avamar	4.9 MB	2.1%

Sch/Disp: Running/Running | No Unacknowledged Events | Server: Full Access

- To filter the results to display only restore activity, select **Actions > Filter**.

The **Filter Activity** dialog box appears.

- In the **Filter Activity** dialog box, select **Restore** from the **Type** list.
- Click **OK**.

Canceling restores

1. In Avamar Administrator, click the **Activity** tab.
The **Activity** window appears.
2. Select the **Activity Monitor** tab.
A list of all activities appears.
3. From the list of activities, click the restore to highlight it.
4. Select **Actions > Cancel Activity**.
A confirmation message appears.
5. Click **Yes**.
The cancellation may take five minutes or more to complete. The restore may complete before the cancellation finishes.

CHAPTER 5

Disaster Recovery

The following topics describe how to prepare for and perform disaster recovery of a SQL Server environment:

- ◆ [Preparing for disaster recovery](#) 152
- ◆ [Performing disaster recovery](#) 152

Preparing for disaster recovery

To prepare for disaster recovery of a SQL Server environment:

1. Ensure that you have onsite and offsite copies of the installation disks for the operating system, SQL Server, and any software updates.
2. Perform full Windows server backups by using the Avamar Client for Windows. Include System State and all critical and noncritical disks in the backups. The *EMC Avamar for Windows Server User Guide* provides full backup instructions.
3. Back up all system and user databases by using the Avamar Plug-in for SQL Server. [Chapter 3, “Backup,”](#) provides backup instructions.

If you use AlwaysOn availability groups, then you must perform a separate backup of the system databases. During the backup, select the server name of the physical node as the client to back up, and then select the system databases. SQL Server does not support system databases in an availability group.

NOTICE

If you perform the full Windows server backup and the database backups at different times, then perform the full Windows server backup first.

Performing disaster recovery

To recover a SQL Server environment after a disaster:

1. Restore the server from the Windows client backup as described in the *EMC Avamar for Windows Server User Guide*.
2. Complete the recommended Microsoft SQL Server disaster recovery procedure by using the steps in the “Recovering from a Disaster” article on the MSDN website.
3. Use the SQL Server plug-in to restore the system databases as described in [“Restoring system databases” on page 132](#).
4. Use the SQL Server plug-in to restore the user databases as described in [“Restoring to the original location” on page 106](#).
5. When you are sure that the restore of the environment is correct and complete, perform a full backup of the newly restored environment.

APPENDIX A

Plug-in Options

The following topics provide information about backup and restore plug-in options for the Avamar Plug-in for SQL Server:

- ◆ [How to set plug-in options 154](#)
- ◆ [Backup options 155](#)
- ◆ [Restore options 158](#)

How to set plug-in options

Plug-in options enable you to control specific actions for on-demand backups, restores, and scheduled backups. The plug-in options that are available depend on the operation type and client plug-in type.

You specify plug-in options in Avamar Administrator for on-demand backup or restore operations, or when you create a dataset for a scheduled backup. You set plug-in options with the graphical user interface (GUI) controls (text boxes, checkboxes, radio buttons, and so forth). In addition to using the GUI controls for the options, you can type an option and its value in the **Enter Attribute** and **Enter Attribute Value** fields in special circumstances.

NOTICE

The Avamar software does not check or validate the information that you type in the **Enter Attribute** and **Enter Attribute Value** fields. In addition, the values in the **Enter Attribute** and **Enter Attribute Value** fields override settings that you specify with the GUI controls for the options.

Detailed instructions on how to access and set plug-in options during a backup or restore are available in [Chapter 3, “Backup,”](#) and [Chapter 4, “Restore.”](#)

Backup options

The following options are available for the Avamar Plug-in for SQL Server when you perform an on-demand backup or when you configure a dataset for scheduled backups. An asterisk (*) indicates an advanced option, which only appears when you select **Show Advanced Options** on the **Backup Command Line Options** dialog box.

Table 13 Backup plug-in options (page 1 of 4)

Setting	Description
Backup label	Assigns this descriptive label to the backup.
Backup type	<p>Sets the backup level. One of the following values:</p> <ul style="list-style-type: none"> • Full — Backs up the entire database, including all objects, system tables, and data. • Differential — Backs up any data that changed after the last complete backup. • Incremental — Backs up only the transaction logs. <p>Notice: Microsoft SQL Server does not allow differential or transaction log (incremental) backups of the master database. You can only create full backups of the master database. If you try to perform a differential or transaction log backup of the master database, a full backup occurs instead.</p>
Availability group replica for backup	<p>Controls the replica on which the backup occurs in an AlwaysOn environment. One of the following values:</p> <ul style="list-style-type: none"> • Primary — Specifies that the backup should occur on the primary replica. • Prefer secondary — Specifies that the backup should occur on a secondary replica except when the primary replica is the only available replica. This is the default value. • Secondary only — Specifies that the backup should occur on the secondary replica. If no secondary replica is available, then the backup does not occur. • SQL Server defined — Specifies that the backup should use the backup preference specified in SQL Server settings. <p>You can increase performance of the primary replica when you perform backups on a secondary replica. However, the secondary replica must meet the requirements in “AlwaysOn availability group requirements” on page 43.</p>

Table 13 Backup plug-in options (page 2 of 4)

Setting	Description
Enhanced data deduplication	<p>Controls the type of compression for data in a backup by setting the <code>--compress</code> option in <code>avtar</code>. Enhanced data deduplication typically reduces the amount of client data that the backup process must send to the server, but requires additional client CPU resources. Select one of the following values:</p> <ul style="list-style-type: none"> • Default — The <code>avtar</code> process evaluates some of the data in the backup to determine whether the data supports compression. If the evaluated data supports compression, then <code>avtar</code> compresses all data in the backup. If the evaluated data does not support compression, then <code>avtar</code> does not compress any data. • Disabled — The <code>avtar</code> process does not compress data during the backup. • Enabled — Forces <code>avtar</code> to compress data during the backup, even if the data does not support compression. Select this value only when you are certain that data in the backup supports compression. Otherwise, <code>avtar</code> unnecessarily uses additional client CPU resources to try to perform compression on data that does not support compression.
Force incremental backup after full backup	<p>If a full backup exists and then another full backup occurs, this option creates a transaction log (incremental) backup. The transaction log backup contains transactions from the end of the first full backup or the most recent transaction log backup, whichever is later, until the time that the forced transaction log backup occurs (after the first backup of the new backup cycle).</p> <p>This ensures that a point-in-time recovery to a point in time between the two full backups can occur, if required.</p> <p>“Force incremental backup after full backup option” on page 81 provides additional details.</p> <hr/> <p>Notice: You cannot perform a transaction log backup on databases that use the simple recovery model because those databases do not support transaction log backups. This includes system databases such as the master and msdb databases. Use the For simple recovery model databases list to control what Avamar does when you try to perform transaction log backups of databases that use the simple recovery model.</p>
*Force full backup	<p>Performs a full backup in the following scenarios:</p> <ul style="list-style-type: none"> • If you try to perform a differential or transaction log (incremental) backup and a full backup does not exist on the server. • If you try to perform a differential or transaction log backup after you restore a database. • If you try to perform a differential or transaction log backup after you restore the msdb database. • If log gaps are detected. <p>Effectively, this option automates taking a full backup when necessary.</p> <p>As a best practice, perform a transaction log backup after the forced full backup.</p> <hr/> <p>Notice: If you perform transaction log and differential backups, EMC strongly recommends that you leave this option selected (the default setting) for all backups. Otherwise, you might not be able to restore data if there are no full backups on the Avamar server.</p>

Table 13 Backup plug-in options (page 3 of 4)

Setting	Description
*Truncate database log	<p>Controls database transaction log truncation behavior. One of the following values:</p> <ul style="list-style-type: none"> • (Default) Only for incremental backup — The backup process truncates the database transaction log if the backup type is incremental (transaction log). No log truncation occurs if the backup type is full or differential. • For all backup types — The backup process truncates the database transaction log regardless of the backup type. This setting breaks the chain of log backups. Do not use this setting unless the backup type is full. This value is not supported for SQL Server 2008 R2 and SQL Server 2012 environments. • Never — The backup process does not truncate the database transaction log under any circumstances.
*For simple recovery model databases	<p>Specifies what Avamar does when you try to perform incremental (transaction log) backups of databases that use the simple recovery model, which does not support transaction log backups:</p> <ul style="list-style-type: none"> • Skip incremental with error — If you select databases with different recovery models for the backup, then the backup does not include the databases with the simple recovery model. The backup completes with exceptions, and an error message is written to the log. If you select only databases with the simple recovery model for the backup, then the backup fails. • Skip incremental with warning — If you select databases with different recovery models for the backup, then the backup does not include databases with the simple recovery model. The backup completes successfully, and a warning is written to the log for each database that uses the simple recovery model. If you select only databases with the simple recovery model for the backup, then the backup fails. • Promote incremental to full — A full backup occurs automatically instead of a transaction log backup for databases that use the simple recovery model.
*Enable debugging messages	Specifies to write maximum information to log files for debugging purposes. The backup process generates very large log files.
Store backup on Data Domain system	<p>Stores the backup on a configured Data Domain system instead of on the Avamar server.</p> <p>To store the backup on a Data Domain system, select the checkbox and then select the Data Domain system from the list.</p>
Multi-streaming options	
Maximum number of streams	<p>Specifies the maximum number of streams to use for backup and restore. The default is 2, and the maximum value is 10.</p> <p>If you select a value of 1, then Avamar backs up one database at a time with one stream.</p> <p>“Multi-streaming options” on page 85 provides additional details.</p>
Minimum stream size	Specifies the minimum size of each data stream.
Authentication options	
*SQL server address	Connect to this hostname or IP address for the computer that is running SQL Server.

Table 13 Backup plug-in options (page 4 of 4)

Setting	Description
*Authentication method	Specifies whether to use Windows (NT) authentication or SQL Server authentication to connect to SQL Server. If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account. “Security requirements” on page 40 provides more information.
*SQL login ID	If you select SQL server authentication in the Authentication method list, connect to the server with this login ID.
*SQL password	If you select SQL server authentication in the Authentication method list, type the password for the SQL login account.

Restore options

The following options are available when you perform a restore using the Avamar Plug-in for SQL Server. An asterisk (*) indicates an advanced option, which only appears when you select **Show Advanced Options** on the **Restore Command Line Options** dialog box.

Table 14 Restore plug-in options (page 1 of 4)

Setting	Description
Use SQL REPLACE option	<p>Adds an SQL <code>WITH REPLACE</code> clause statement to the restore Transact-SQL command. The statement specifies that SQL Server should create any necessary database and related files even if another database or file already exists with the same name.</p> <p>Notice: This option overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The <i>Microsoft Transact-SQL Reference Manual</i> provides more information in the RESTORE command section.</p> <p>You may need to use this option if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log: <code>One or more devices or files already exist. Reissue the statement using the WITH REPLACE option to overwrite these files and devices.</code></p> <p>You should also use this option when restoring system databases.</p>
Tail-log backup	<p>Backs up the tail of the transaction log to capture the log records that have not been backed up. The restore process uses the tail-log backup after the database restore to recover the transactions that were not in the backup.</p> <p>Tail-log backup is available only for databases that use the full and bulk-logged recovery models. Perform a tail-log backup during a direct restore without the SQL REPLACE option.</p> <p>Clear the checkbox when you perform a redirected restore to a different SQL Server instance.</p>
Restore only on primary replica	<p>Restores the database on only the primary replica in an Always On availability group. When you select this option, you must manually perform database synchronization on the secondary replicas.</p> <p>“Restoring to an AlwaysOn availability group” on page 138 provides details on restoring to an AlwaysOn availability group.</p>

Table 14 Restore plug-in options (page 2 of 4)

Setting	Description
*Recovery operation	<p>Specifies the recovery operation that occurs after the restore:</p> <ul style="list-style-type: none"> • RECOVERY — The database is fully recovered and online. This is the default setting. • NORECOVERY — The database remains in a restoring state. This enables you to perform additional manual restore tasks, such as applying additional SQL log files. • STANDBY — The database is in standby (read-only) mode. This option creates a file with recovery changes. You can use the file to revert the recovery changes, if required. The size of the file depends on the volume of undo actions from uncommitted transactions. Specify the path to this file in the Standby file location box. STANDBY enables you to bring up a database for read-only access between transaction log restores. Use this setting with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores. Do not use STANDBY when a database upgrade is necessary. You may need to perform a database upgrade when restoring backup sets from an earlier version of SQL Server.
*Standby file location	<p>When you select STANDBY from the Recovery operation list, use this setting to specify the path to the standby file with recovery changes.</p> <p>The path must be in the following format:</p> <p><i>drive:\folder\subfolder</i></p> <p>where <i>drive</i> is the drive letter on the destination client, and <i>folder\subfolder</i> is the path on the drive letter in which to create the file.</p> <p>If a file with recovery changes already exists in the location, then SQL Server overwrites it. If you do not specify a path, then Avamar creates the file in the <i>C:\Program Files\avs\var</i> folder, where <i>C:\Program Files\avs</i> is the Avamar installation folder.</p>
*Enable debugging messages	Specifies to write maximum information to log files for debugging purposes. The restore process generates very large log files.
System Databases options	
Restore system databases	Causes the plug-in to create an expanded list of databases to restore that includes system databases and user databases. By default, the plug-in only shows user databases.
Manage SQL services automatically during restore	<p>When you restore the master database, this option automatically stops the SQL Server instance, including dependent services such as the SQL Server agent service and the Analysis Service, and restarts the instance in single-user mode before the restore. After the restore, the instance automatically restarts.</p> <p>When you restore the msdb database, this option automatically stops the SQL Server agent service, and then restarts it when the restore is complete.</p> <p>When you select both system and user databases for restore, you must select this option to ensure that all system databases restore in the proper order and with the necessary service stops and restarts.</p>

Table 14 Restore plug-in options (page 3 of 4)

Setting	Description
Redirected Restore options	
*New database name	To give a database a different name when you restore it to either a different instance on the same server or to a different server, type the new name here. Only the name of the restored database is different. The source database in the backup remains the same.
Alternate database location	If you are relocating the database, type the full path of the new database file (.mdf) location on the client, such as C:\temp. Notice: You also can specify the file locations by clicking Set Destination on the Restore Options dialog box. If you specify the file locations in the Restore Options dialog box, then you do not need to specify them as plug-in options on the Restore Command Line Options dialog box. If you are restoring a database with multiple filegroups or files and you want to restore the files to different locations, then you <i>must</i> use the redirected restore options on the Restore Options dialog box.
Alternate log location	Specifies whether the database log file (.ldf) restores to the same location as the database or to a different location.
*Path to alternate log location	If you are restoring the database log file to a different location than the database, type the full path of the new database log file location on the client, such as C:\temp\logs.
Authentication options	
*SQL server address	Connect to this hostname or IP address for the computer that is running SQL Server
*Authentication method	Specifies whether to use Windows (NT) authentication or SQL Server authentication to connect to SQL Server. If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account. “Security requirements” on page 40 provides more information.
*SQL login ID	If you select SQL server authentication in the Authentication method list, connect to the server with this login ID.
*SQL password	If you select SQL server authentication in the Authentication method list, type the password for the SQL login account.
Point-in-Time Recovery options	
*Point-in-time recovery mode	Specifies whether to recover the database to a specific date and time or a named mark in the transaction log. Available only if the database uses the full recovery model.
*Point-in-time or mark name string	If you select Point-in-time or Mark name from the Point-in-time recovery mode list, then this option specifies the point in time or named mark to which to recover: <ul style="list-style-type: none"> Point-in-time — To recover to a specific point in time, specify the date and time in yyyy-mm-ddThh:mm:ss format. For example, 2013-10-25T14:15:45 is October 25, 2013 at 2:15:45 p.m. Mark name — If named marks are inserted in the transaction log for recovery purposes, then you can recover to a specific mark. Specify the mark name.

Table 14 Restore plug-in options (page 4 of 4)

Setting	Description
*Mark recovery point	Specifies whether to include the mark in the recovery: <ul style="list-style-type: none"> • Before mark — Specifies that the log record immediately before the mark is the recovery point. In other words, the recovery rolls forward to the mark and excludes the marked transaction. • At mark — Specifies that the marked transaction is the recovery point. In other words, the recovery rolls forward to the mark and includes the marked transaction.
*Mark is after date/time	If named marks are not unique in the transaction log, then you can specify a date and time to locate the mark to which to recover. The recovery process stops at the first mark with the specified name, exactly at or after the specified date and time. Specify the date and time in yyyy-mm-ddThh:mm:ss format. For example, 2013-10-25T14:15:45 is October 25, 2013 at 2:15:45 p.m.

APPENDIX B

Command Line Interface

The following topics explain how to use the command line interface (CLI) for the Avamar Plug-in for SQL Server to back up or restore SQL Server data:

- ◆ [Understanding the SQL Server plug-in CLI](#) 164
- ◆ [Command reference](#) 168
- ◆ [CLI examples](#) 180

Understanding the SQL Server plug-in CLI

To understand the SQL Server plug-in CLI, review the architecture, launch steps, available operations and options, and other details.

CLI architecture

When you use the CLI to initiate a backup or restore, you specify the options for the SQL Server plug-in binary on the command line. The plug-in interacts with the `avtar` process to write backup data to or read backup data from the Avamar server.

When you use Avamar Administrator instead, the Management Console Server (MCS) service on the Avamar server gives the `avagent` process on the client a workorder with the options that you specify through Avamar Administrator. The `avagent` process then starts the SQL Server plug-in, which then interacts with `avtar` for the backup or restore.

With the CLI, the MCS and `avagent` process are not involved in the backup or restore.

Launching the CLI

The `avsql` binary is located in `C:\Program Files\avs\bin`, where `C:\Program Files\avs` is the Avamar client installation folder. To launch the CLI, open a command prompt and change directory to the bin folder of the Avamar client installation folder. [“Command reference” on page 168](#) provides a complete list of available commands and options.

Available operations

The following topics provide an overview of the operations that are available with the `avsql` command.

To specify the operation, use `--operation={browse | backup | restore}` on the `avsql` command line.

Browse

The `browse` operation returns a list of data on the client that is available for backup. You can browse all data on the client, a single instance, or a single database.

The command results appear as standard output in the command window. The following information appears for each entry:

- ◆ Name
- ◆ Date
- ◆ Size
- ◆ Type
- ◆ User

The data sorts alphabetically by name.

Backup

The `backup` operation performs an on-demand backup of the specified data.

You can specify plug-in options for the backup. Many of these plug-in options are the same options that you specify when you perform an on-demand backup or configure a scheduled backup in Avamar Administrator.

The **Enhanced data deduplication** plug-in option appears in the **Backup Command Line Options** dialog box but is not available as an option when you perform a backup with the CLI.

Restore

The `restore` operation restores the specified data from either the Avamar server or a Data Domain system, depending on where the backup is stored.

You can specify plug-in options for the restore. Many of these plug-in options are the same options that you specify when you perform a restore in Avamar Administrator. The **Manage SQL services automatically during restore** option is the only plug-in option that appears in the **Restore Command Line Options** dialog box but is not available when you perform a restore with the CLI.

Options

You can specify options for `avsql` to control backup or restore behavior.

There are several ways to specify options for the `avsql` command:

- ◆ Type the individual options on the command line.
- ◆ List the options in the `avsql.cmd` file, which is located in the `C:\Program Files\avs\var` folder, where `C:\Program Files\avs` is the Avamar client installation folder. List each option on its own line, as shown in the following example:

```
--debug
--logfile=avamarclient.log
--verbose=5
```

- ◆ Create an option file as a text file, and then specify the option file on the command line using the `--flagfile=path` option, where *path* is the full path and file name of the option file.
- ◆ You can pass options to the CLI from environment variables. For example, if you specify `Instance1=NamedInstance1` in the environment variables, then you can browse the instance by typing the following command:

```
avsql --operation=browse %Instance1%
```

Using the CLI in a failover cluster

If you are using the CLI to browse, back up, or restore SQL Server data on shared storage in a failover cluster, then the following steps are required:

- ◆ Run the CLI on the on the active node that is managing the databases to back up.
- ◆ Specify the cluster client as the client to back up or restore from.
- ◆ To browse, back up, or restore the local instance, specify the `--sqlserver` option with a value of the cluster client.

Using the CLI in an AlwaysOn availability group

If you are using the CLI to browse, back up, or restore SQL Server data in an AlwaysOn availability group, then the following steps are required:

- ◆ Run the CLI on the cluster node with the primary replica.
- ◆ Specify the cluster client for the availability group listener as the client to back up or restore from.
- ◆ During a backup, specify the following options:
 - Specify the availability group listener with the `--sqlserver` and `--hostname` options.
 - Specify the availability group listener, the availability group, the instance, and the database in quotation marks using the format `"listener/group\instance\database"`. To back up all databases in an availability group, specify `"listener/group\instance"`.
- ◆ When you restore a backup to its original location, specify the following options:
 - Specify the availability group listener in the `--sqlserver` and `--hostname` options.
 - Use the `--recovery-primary-only` option to control whether to restore to only the primary replica or to both the primary and all secondary replicas.
- ◆ When you restore a backup to its original location but with a new name, specify the original database name at the end of the command, and use the `--newdbname` option to specify the new database name.

Password encoding

You can use `avtar` to encode passwords that you type on the command line or store in script files, and then use the encoded string with the `--password`, `--ap`, or `--pswd` option.

To encode passwords:

1. Type the following command on the command line on the SQL Server client:

```
avtar --encodepassword=password
```

where *password* is the password to encode.

An encoded string is returned on the command line.

2. Use the encoded string on the `avsql` command line as the password with the `--password`, `--ap`, or `--pswd` option.

Help

To view command line help output, use the `--help` option. The following command shows a complete list of available operations and options, with a description for each one:

```
avsql --help
```

To view only the options for a specific operation, use the `--help` and `--operation=operation` options together. The following command provides a list of options for the backup command:

```
avsql --help --operation=backup
```

Monitoring CLI activities

The status of backups and restores that you perform by using the SQL Server plug-in CLI appears on the **Session Monitor** tab in the **Server** window in Avamar Administrator.

CLI activities do not appear in the **Activity Monitor** tab in the **Activity** window. The **Activity Monitor** displays only backups and restores that you perform by using Avamar Administrator.

Command reference

The `avsql` command enables you to browse for SQL Server data to back up or restore, perform the backup, or perform the restore.

Synopsis

```
avsql --operation={browse | backup | restore} [options]
```

Operations

Supply one and only one of the following operations for `avsql` using the `--operation` option.

Table 15 Operations for the `avsql` command

Operation	Description
<code>browse</code>	Returns a list of data on the client that is available for backup. You can browse all data on the client, a single instance, or a single database. You must specify only a single browse target on the command line. If you specify more than one instance or database, then <code>avsql</code> displays output for only the first target.
<code>backup</code>	Performs an on-demand backup of the specified data.
<code>restore</code>	Restores the specified data.

Options

The following topics list the options that are available for `avsql`.

Common options

The following common options are available for `avsql`.

Table 16 Common `avsql` options

Option	Description
<code>--backup-type=type</code>	Internal use only.
<code>--exclude-pluginid-list=string</code> <code>--excludepluginids=string</code>	Internal use only.
<code>--pluginid-list=string</code> <code>--pluginids=string</code>	Internal use only.
<code>--version</code>	Displays the build version of the Avamar Plug-in for SQL Server.
<code>--sqltimeoutsecs=seconds</code>	The number of seconds after which the connection to SQL Server times out. If not specified, then the timeout occurs after 100 seconds.

Account options

The following account options are available for `avsql`.

Table 17 Account options for the `avsql` command (page 1 of 2)

Option	Description
<code>--account=domain/client</code> <code>--path=domain/client</code> <code>--acnt=domain/client</code>	Required for all backup and restore operations. Specifies the client to back up or restore from using the following format: <i>domain/client</i> where <i>domain</i> is the Avamar domain to which the client belongs and <i>client</i> is the name of the client computer.
<code>--hostname=client</code>	Specifies the name of the client to back up or restore from. Required for backups and restores in an AlwaysOn availability group. Specify the AlwaysOn availability group listener as the client.
<code>--id=user@domain/client</code>	Required for all backup and restore operations. Specifies the Avamar username for authentication. Note: You can specify the values for the <code>--id</code> and <code>--account</code> options simultaneously by specifying <code>--id=user@domain/client</code> .
<code>--password=password</code> <code>--ap=password</code> <code>--pswd=password</code>	Required for all backup and restore operations. Specifies the password for the Avamar account.
<code>--server=Avamar_server</code> <code>--hfsaddr=Avamar_server</code>	Required for all backup and restore operations. Specifies the hostname or IP address of the Avamar server.
<code>--ntorsqlauth={ntauthentication sqlauthentication}</code>	Specifies whether to use Windows (NT) authentication or SQL Server authentication to connect to SQL Server. One of the following values: <ul style="list-style-type: none"> <code>ntauthentication</code> <code>sqlauthentication</code> If you do not specify an authentication method, then the SQL Server plug-in uses NT authentication and logs in with the NT AUTHORITY\SYSTEM account. “ Security requirements ” on page 40 provides more information.
<code>--pluginport=port</code>	Specifies the port number on which to connect to the agent. If not specified, then the default value of 28002 is used.

Table 17 Account options for the `avsql` command (page 2 of 2)

Option	Description
<code>--sqllogin=name</code>	If you set <code>--ntorsqlauth=sqlauthentication</code> , then this option specifies the login ID for the SQL login account.
<code>--sqlpassword=password</code>	If you set <code>--ntorsqlauth=sqlauthentication</code> , then this option specifies the password for the SQL login account.
<code>--sqlserver=SQLserver</code>	<p>Hostname or IP address of the computer with the SQL Server installation that you want to browse, back up, or restore from.</p> <p>This option is required when you use the CLI to browse, back up, or restore data in the following environments:</p> <ul style="list-style-type: none"> • In the local instance when the server is in a failover cluster. Specify the virtual server name. • In an AlwaysOn availability group. Specify the availability group listener.

Logging options

The following logging options are available for `avsql`.

Table 18 Logging options for the `avsql` command

Option	Description
<code>--informationals=n</code>	Sets the information level for status messages, where <i>n</i> is a number such as 0, 1, 2, and so on.
<code>--log=file --logfile=file</code>	<p>Specifies the full path and file name of the <code>avsql</code> plug-in log file.</p> <p>Notice: The Activity Monitor in Avamar Administrator displays only <code>avtar</code> logs for backups with the command line interface. To create a plug-in log for CLI activities, you must specify the <code>--log</code> or <code>--logfile</code> option on the command line.</p>
<code>--noinformationals</code>	Disables all status messages.
<code>--nostdout</code>	Disables output to STDOUT. However, if you supply <code>--log</code> and <code>--logfile=file</code> , then output still goes to the log file.
<code>--nowarnings</code>	Disables warning messages.
<code>--quiet</code>	Suppresses all debugging messages.
<code>--verbose -v</code>	Enables all messages, including status and warning messages.

Browse options

There are no browse options for `avsql`.

Backup options

The following `avsql` options are available for the `backup` operation.

Table 19 Backup options for the `avsql` command (page 1 of 4)

Option	Description
<code>--ddr={true false}</code>	Specifies whether to store the backup on a Data Domain system instead of the Avamar server. To store the backup on a Data Domain system, specify <code>true</code> .
<code>--ddr-index=<i>n</i></code>	Specifies the index number (1, 2, 3, and so forth) of the Data Domain system on which to store the backup. The index number is assigned to the Data Domain system when it is added to the Avamar server configuration.
<code>--encrypt=<i>value</i></code>	Specifies the encryption method for client/server data transfer during the backup. One of the following values: <ul style="list-style-type: none"> • <code>proprietary</code> • <code>tcp</code> • <code>ssl</code> • <code>sslverify</code> • <code>tls</code> • <code>tls-sa</code> Use this option with the <code>--encrypt-strength</code> option. The <i>EMC Avamar Product Security Guide</i> provides details on how to correctly specify the <code>--encrypt</code> and <code>--encrypt-strength</code> option values together to ensure proper encryption settings.
<code>--encrypt-strength=<i>value</i></code>	Specifies the encryption strength for client/server data transfer during the backup. One of the following values: <ul style="list-style-type: none"> • <code>cleartext</code> • <code>medium</code> • <code>high</code> Use this option with the <code>--encrypt</code> option. The <i>EMC Avamar Product Security Guide</i> provides details on how to correctly specify the <code>--encrypt</code> and <code>--encrypt-strength</code> option values together to ensure proper encryption settings.
<code>--expires={<i>days</i> <i>timestamp</i>}</code>	Specifies backup expiration as a number of days from today (<i>days</i>) or an absolute <i>timestamp</i> .
<code>--label=<i>name</i></code>	Assigns this descriptive label to the backup.
<code>--retention-type=<i>type</i></code> <code>--retentiontype=<i>type</i></code>	Assigns advanced retention to the backup. One of the following values: <ul style="list-style-type: none"> • <code>none</code> — Do not explicitly assign any retention type to this backup. That is, treat the backup as a normal on-demand backup. • <code>daily</code> — Explicitly designate this backup as a daily backup. • <code>weekly</code> — Explicitly designate this backup as a weekly backup. • <code>monthly</code> — Explicitly designate this backup as a monthly backup. • <code>yearly</code> — Explicitly designate this backup as a yearly backup.
<code>--after-date=<i>date</i></code>	To improve the performance of incremental backups, this option specifies the date at which a week-by-week search for a full backup stops and the history of all backups is retrieved. Specify the date in <code>mm/dd/yyyy</code> format.

Table 19 Backup options for the `avsql` command (page 2 of 4)

Option	Description
<pre>--backup-preference= {primary prefer_secondary secondary_only SQL_server_defined}</pre>	<p>Controls the replica on which the backup occurs in an AlwaysOn environment. One of the following values:</p> <ul style="list-style-type: none"> <code>primary</code> — Specifies that the backup should occur on the primary replica. <code>prefer_secondary</code> — Specifies that the backup should occur on a secondary replica except when the primary replica is the only available replica. This is the default value. <code>secondary_only</code> — Specifies that the backup should occur on the secondary replica. If no secondary replica is available, then the backup does not occur. <code>SQL_server_defined</code> — Specifies that the backup should use the backup preference specified in SQL Server settings. <p>You can increase performance of the primary replica when you perform backups on a secondary replica. However, the secondary replica must meet the requirements in “AlwaysOn availability group requirements” on page 43.</p>
<pre>--brtype={full differential incremental}</pre>	<p>Specifies the type of backup to perform.</p> <ul style="list-style-type: none"> <code>full</code> — Backs up the entire database, including all objects, system tables, and data. <code>differential</code> — Backs up any data that has changed since the last complete backup. <code>incremental</code> — Backs up only the transaction logs. <p>The default value is <code>full</code>.</p> <p>Notice: Microsoft SQL Server does not allow differential or transaction log (incremental) backups of the master database. You can only create full backups of the master database. If you try to perform a differential or transaction log backup of the master database, a full backup is created instead.</p>
<pre>--exclude=<i>data</i></pre>	<p>Specifies an instance or database to exclude from the backup. When specifying the local instance, use <code>(local)</code>. For example:</p> <ul style="list-style-type: none"> <code>--exclude=(local)</code> <code>--exclude=(local)/DB1</code>
<pre>--forcefull</pre>	<p>Performs a full backup in the following scenarios:</p> <ul style="list-style-type: none"> If you try to perform a differential or transaction log (incremental) backup and a full backup does not exist on the server. If you try to perform a differential or transaction log backup after you restore a database. If you try to perform a differential or transaction log backup after you restore the msdb database. If log gaps are detected. <p>Effectively, this option automates taking a full backup when necessary.</p> <p>Notice: If you perform transaction log and differential backups, EMC strongly recommends that you leave the default setting of enabled for the <code>--forcefull</code> option for all backups. Otherwise, you might not be able to restore data if no existing full backup is present on the Avamar server.</p>

Table 19 Backup options for the `avsql` command (page 3 of 4)

Option	Description
<pre>--forceincremental={true false}</pre>	<p>If a full backup exists and then another full backup occurs, this option creates a transaction log (incremental) backup that contains transactions that occur from the end of the first full backup or the most recent transaction log backup, whichever is later, until the time that the forced transaction log backup occurs (after the first backup of the new backup cycle). This ensures that a point-in-time recovery to a point in time between the two full backups can occur, if required.</p> <p>Three backup sessions occur when a forced transaction log backup occurs:</p> <ul style="list-style-type: none"> • The first session contains transactions that occur from the end of the first full backup until the start time of the forced transaction log backup. • The second session contains transactions that occur from the end of the second full backup until the time that the forced transaction log backup occurs. • The third session contains only the full backup. <p>To restore and recover the database to a point after the forced transaction log backup, select the last backup from the last backup session that has the most recent full backup, during the restore process.</p> <hr/> <p>Notice: You cannot perform a transaction log backup on databases that use the simple recovery model because those databases do not support transaction log backups. This includes system databases such as the master and msdb databases. As a result, set <code>--forceincremental=false</code> when you back up a database that uses the simple recovery model.</p> <hr/>
<pre>--include=data</pre>	<p>Specifies an instance or database to include in the backup that otherwise would have been excluded based on the <code>--exclude</code> option.</p> <p>When specifying the local instance, use <code>(local)</code>. For example:</p> <ul style="list-style-type: none"> • <code>--include=(local)</code> • <code>--include=(local)/DB1</code>
<pre>--max-parallel=n</pre>	<p>Specifies the maximum number of streams to use for backup and restore. The default is 1, and the maximum value is 10. “Multi-streaming options” on page 85 provides details.</p>
<pre>--min-snapup-size=MB</pre>	<p>Specifies the minimum size of each data stream in MB. The default value is 200.</p>

Table 19 Backup options for the `avsql` command (page 4 of 4)

Option	Description
<code>--prefix=value</code>	Specifies a prefix for the log filename.
<code>--skip_or_prom_sdm={skip_with_error skip_with_warning promote_to_full}</code>	<p>Specifies what Avamar does when you try to perform incremental (transaction log) backups of databases that use the simple recovery model, which does not support transaction log backups:</p> <ul style="list-style-type: none"> <code>skip_with_error</code> — If you select databases with different recovery models for the backup, then the backup does not include the databases with the simple recovery model. The backup completes with exceptions, and an error message is written to the log. If you select only databases with the simple recovery model for the backup, then the backup fails. <code>skip_with_warning</code> — If you select databases with different recovery models for the backup, then the backup does not include databases with the simple recovery model. The backup completes successfully, and a warning is written to the log for each database that uses the simple recovery model. If you select only databases with the simple recovery model for the backup, then the backup fails. <code>promote_to_full</code> — A full backup occurs automatically instead of a transaction log backup for databases that use the simple recovery model. <p>The default value is <code>skip_with_error</code>.</p>
<code>--truncatelog_ts={default enable disable}</code>	<p>Specifies database transaction log truncation behavior. One of the following values:</p> <ul style="list-style-type: none"> <code>default</code> — The backup process truncates the database transaction log if the backup type is set to incremental (transaction log). No log truncation occurs if the backup type is full or differential. <code>enable</code> — The backup process truncates the database transaction log regardless of the backup type. This setting breaks the chain of log backups and should not be used unless the backup type is set to full. This value is not supported for SQL Server 2008 R2 and SQL Server 2012 environments. <code>disable</code> — The backup process does not truncate the database transaction log under any circumstances.

Restore options

The following `avsql` options are available for the restore operation.

Table 20 Restore options for the `avsql` command (page 1 of 6)

Option	Description
<code>--label=name</code>	Specifies the label of the backup to restore. Use either this option or the <code>--labelnum</code> option during the restore.
<code>--labelnum=number</code>	Required for all restores. Specifies the label number of the backup to restore.
<code>--after-date=date</code>	To improve the performance of the search for backups to restore, this option specifies the date at which a week-by-week search for a full backup stops and the history of all backups is retrieved. Specify the date in <code>mm/dd/yyyy</code> format.

Table 20 Restore options for the `avsql` command (page 2 of 6)

Option	Description
<code>--aftertime=datetime</code>	If you are performing a point-in-time recovery to a named mark and named marks are not unique in the transaction log, then you can use this option to specify a date and time to locate the mark to which to recover. The recovery process stops at the first mark with the specified name, exactly at or after the specified date and time. Specify the date and time in <code>yyyy-mm-ddThh:mm:ss</code> format. For example, <code>2013-10-25T14:15:45</code> is October 25, 2013 at 2:15:45 p.m.
<code>--altlog={samelocat difflocat}</code>	Specifies whether the database log file (<code>*.ldf</code>) is restored to the same location as the database or to a different location. One of the following values: <ul style="list-style-type: none"> <code>samelocat</code> <code>difflocat</code> The default value is <code>samelocat</code> .
<code>--app-instance=instance</code>	When restoring to a different instance, this option specifies the target instance for the restore.
<code>--dblocation=path</code>	If you are relocating the database, this option specifies the full path of the new database file (<code>*.mdf</code>) location on the client, where <code>path</code> is a value such as <code>C:\temp</code> .
<code>--encrypt=value</code>	Specifies the encryption method for client/server data transfer during the restore. One of the following values: <ul style="list-style-type: none"> <code>proprietary</code> <code>tcp</code> <code>ssl</code> <code>sslverify</code> <code>tls</code> <code>tls-sa</code> Use this option with the <code>--encrypt-strength</code> option. The <i>EMC Avamar Product Security Guide</i> provides details on how to correctly specify the <code>--encrypt</code> and <code>--encrypt-strength</code> values together to ensure specific encryption settings.
<code>--encrypt-strength=value</code>	Specifies the encryption strength for client/server data transfer during the restore. One of the following values: <ul style="list-style-type: none"> <code>cleartext</code> <code>medium</code> <code>high</code> Use this option with the <code>--encrypt</code> option. The <i>EMC Avamar Product Security Guide</i> provides details on how to correctly specify the <code>--encrypt</code> and <code>--encrypt-strength</code> values together to ensure specific encryption settings.
<code>--handle-sql-services-automatically</code> <code>--handlesqlservicesautomatically</code>	When restoring system databases, this option stops and restarts SQL services automatically as necessary. When you restore both system and user databases, you must specify this option to ensure that all system databases restore in the proper order and with the necessary service stops and restarts.

Table 20 Restore options for the `avsql` command (page 3 of 6)

Option	Description
<code>--history-optimization={true false}</code>	Enables or disables history search optimization, which you can further refine by using the <code>--after-date=date</code> option. The default setting is <code>--history-optimization=true</code> , which enables history search optimization.
<code>--loglocation=path</code>	If <code>--altlog=difflocat</code> , then this option specifies the path of the new database log file location on the client, where <i>path</i> is a value such as <code>C:\temp\logs</code> .
<code>--logtail</code>	Backs up the tail of the transaction log to capture the log records that are not in a backup. After the database restore, Avamar uses the tail-log backup to recover the transactions that were not in the backup. Tail-log backup is available only for databases that use the full and bulk-logged recovery models. Perform a tail-log backup during a direct restore without the <code>SQL REPLACE</code> option. Do not use this option when you restore to a different SQL Server instance.
<code>--metadata</code>	Retrieves the SQL metadata file from the backup and displays the metadata as screen output. When you use this option, Avamar does not restore any data.
<code>--newdbname=name</code>	Restores the database with the specified name (instead of its original name) when you are restoring it to either a different instance on the same server or to a different server.
<code>--pitormark={none pointintime markname}</code>	Whether to recover the database to a specific date and time or a named mark in the transaction log. Available only if the database uses the full recovery model. One of the following values: <ul style="list-style-type: none"> <code>none</code> — Do not recover to a specific point in time. <code>pointintime</code> — Recover to a specific date and time specified by <code>--pitormarkstr=value</code>. <code>markname</code> — Recover to the named mark specified by <code>--pitormarkstr=value</code>. The default value is <code>none</code> .
<code>--pitormarkstr={pit mark}</code>	If <code>--pitormark=pointintime</code> or <code>markname</code> , then this option specifies the <i>pit</i> (date and time) or named <i>mark</i> to recover to. To recover to a date and time, use <code>yyyy-mm-ddThh:mm:ss</code> format. For example, <code>2013-10-25T14:15:45</code> is October 25, 2013 at 2:15:45 p.m.
<code>--prefix=value</code>	Prefix for the name of the logfiles.
<code>--recovery-primary-only={true false}</code>	Specify <code>true</code> to restore the database on only the primary replica in an Always On availability group. When you specify <code>true</code> , you must manually perform database synchronization on the secondary replicas. Specify <code>false</code> to restore to both primary and secondary replicas.

Table 20 Restore options for the `avsql` command (page 4 of 6)

Option	Description
<pre>--recoveryoperation= {RECOVERY NORECOVERY STANDBY}</pre>	<p>Specifies the recovery operation that occurs after the restore. One of the following values:</p> <ul style="list-style-type: none"> • <code>RECOVERY</code> — The database is fully recovered and online. This is the default setting. • <code>NORECOVERY</code> — The database remains in a restoring state. This enables you to perform additional manual restore tasks, such as applying additional SQL log files. • <code>STANDBY</code> — The database is in standby (read-only) mode. <p>This option creates a file with recovery changes. You can use the file to revert the recovery changes, if required. The size of the file depends on the volume of undo actions from uncommitted transactions. Specify the path to this file with the <code>--standbyfilelocation=</code><i>path</i> option.</p> <p><code>STANDBY</code> enables you to bring up a database for read-only access between transaction log restores. Use this setting with either warm standby server situations or special recovery situations in which it is useful to inspect the database between log restores.</p> <p>Do not use <code>STANDBY</code> when a database upgrade is necessary. You may need to perform a database upgrade when you restore backup sets from an earlier version of SQL Server.</p>
<pre>--recoverypoint= {beforemark atmark}</pre>	<p>When you perform a point-in-time restore to a named mark, this option specifies whether to include the mark in the recovery. One of the following values:</p> <ul style="list-style-type: none"> • <code>beforemark</code> — Specifies that the log record immediately before the mark is the recovery point. In other words, the recovery rolls forward to the mark and excludes the marked transaction. • <code>atmark</code> — Specifies that the marked transaction is the recovery point. In other words, the recovery rolls forward to the mark and includes the marked transaction. <p>The default value is <code>atmark</code>.</p>
<pre>--redirect=<i>string</i></pre>	<p>Restores database files to the specified location, where <i>string</i> contains the original file name and the destination path and file name:</p> <pre>--redirect=<i>original_file</i>= <i>destination_path</i>\<i>new_file</i></pre> <p>To restore multiple files to multiple different locations, separate the <i>string</i> values with commas:</p> <pre>--redirect=<i>original_file1</i>= <i>destination_path1</i>\<i>new_file1</i>, <i>original_file2</i>=<i>destination_path2</i>\ <i>new_file2</i>,<i>original_file3</i>= <i>destination_path3</i>\<i>new_file3</i></pre> <p>To restore a file to its original location, specify only the original filename:</p> <pre>--redirect=<i>original_file</i></pre>
<pre>--redirecttofile={true false}</pre>	<p>Specifies whether to restore database files as operating system files to the specified location.</p>

Table 20 Restore options for the `avsql` command (page 5 of 6)

Option	Description
<code>--restore-destination={original multiple single}</code>	Specifies the restore destination for the data: <ul style="list-style-type: none"> • <code>original</code> – Restore the data to its original location. This is the default value. • <code>multiple</code> – Restore the data to a different location. Use the <code>--target={path client\instance}</code> option to specify the location. • <code>single</code> – Restore the data to operating system files. Use the <code>--target={path client\instance}</code> option to specify the location.
<code>--restore-filegroup-list=filegroups</code>	Restores the specified filegroups, using the following format: <code>--restore-filegroup-list=instance/database/filegroup,instance/database/filegroup</code> where <i>instance/database/filegroup</i> are the instance, database, and filegroup to restore, with multiple entries separated by a comma.
<code>--restoresystem</code>	Attempts to restore databases that have the system attribute set.
<code>--standbyfilelocation=path</code>	If <code>--recoveryoperation=STANDBY</code> , then this option specifies the path on the destination client to the standby file with recovery changes. Use the following format: <code>drive:\folder\subfolder</code> where <i>drive</i> is the drive letter on the destination client, and <i>folder\subfolder</i> is the path on the drive letter in which to create the file. If a file with recovery changes already exists in the specified location, then SQL Server overwrites it. You can use the file to revert the recovery changes, if required.

Table 20 Restore options for the `avsql` command (page 6 of 6)

Option	Description
<pre>--target={path client\instance}</pre>	<p>Specifies the target location for the restored files:</p> <ul style="list-style-type: none"> • If <code>--restore-destination=single</code>, then specify the <i>path</i> on the original client to which to restore the files, such as <code>--target=C:\sqldata</code>. • If <code>--restore-destination=multiple</code>, then specify the <i>instance</i> on the destination client to which to restore the files. Use the following format: <pre>--target=client\instance</pre> where <i>client</i> is the name of the destination client and <i>instance</i> is the name of the instance on the client.
<pre>--use-sql-replace-option --usesqlreplaceoption</pre>	<p>Adds an SQL <code>WITH REPLACE</code> clause statement to the restore Transact-SQL command. This statement specifies that SQL Server should create any necessary database and related files even if another database or file already exists with the same name.</p> <hr/> <p>Notice: This option overrides a SQL Server safety check to prevent you from accidentally overwriting a different database or file. The <i>Microsoft Transact-SQL Reference Manual</i> provides more information in the <code>RESTORE</code> command section.</p> <hr/> <p>You may need to use this option if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log:</p> <pre>One or more devices or files already exist. Reissue the statement using the WITH REPLACE option to overwrite these files and devices.</pre> <p>Use this option when restoring system databases.</p>
<pre>--validate</pre>	<p>Validates the integrity of a backup before performing the restore.</p>

CLI examples

The following topics provide examples of `avsql` commands.

Example browse commands

The following command returns a list of all SQL Server instances on the client:

```
avsql --operation=browse
```

The following command returns a list of all databases in the local instance on the client:

```
avsql --operation=browse (local)
```

The following command returns a list of all databases in the named instance of `Instance1` on the client:

```
avsql --operation=browse Instance1
```

The following command returns detailed information about the database, including the creation date, size, and owner for the database:

```
avsql --operation=browse (local)/DB1
```

Example backup commands

The following topics provide example backup commands.

Backing up all SQL Server data

To back up all SQL Server data (all databases in all instances on the client), specify the domain and client name, and then leave the backup target as a blank space between the quotation marks (" ") at the end of the command.

The following command performs a full backup of all SQL Server data on the `SQLServer1` client in the `SQL` domain, labels the backup `SQL1Full`, and stores the backup on the Avamar server with an IP address of `12.34.56.78`:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin  
--ap=password --brtype=full --path=/SQL/SQLServer1  
--log=C:\Program Files\avs\var\avsql.log " " --label=SQL1Full
```

The following command performs an incremental (transaction log) backup of all SQL Server data on the `SQLServer2` client in the `SQL` domain, labels the backup `SQL2Logs`, and stores the backup on the first Data Domain system that you added to the Avamar configuration:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin  
--ap=password --brtype=incremental --path=/SQL/SQLServer2  
--log=C:\Program Files\avs\var\avsql.log --ddr=true --ddr-index=1  
" " --label=SQL2Logs
```

Backing up the local instance

To back up all databases in the local instance, specify `(local)` as the backup target between the quotation marks at the end of the command.

The following command performs a differential backup of the local instance on the `SQLServer3` client in the `SQL` domain and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=differential --path=/SQL/SQLServer3
--log=C:\Program Files\avs\var\avsql.log "(local)"
```

Backing up a named instance

To back up a named instance, specify the client and instance as the backup target between the quotation marks at the end of the command, using the format `"client\instance"`.

The following command performs a full backup of the `sql02inst2` instance on the `SQLServer2` client in the `SQL` domain and stores the backup on the second Data Domain system that was added to the Avamar configuration:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer2
--log=C:\Program Files\avs\var\avsql.log --ddr=true --ddr-index=2
"SQLServer2\sql02inst2"
```

Backing up multiple instances

To back up one or more instances on a client, but not all instances, specify the instances as separate backup targets between quotation marks at the end of the command.

The following command performs an incremental backup of the local instance and the `sql03inst2` instance on the `SQLServer3` client in the `SQL` domain and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap="password" --brtype=incremental --path="/SQL/SQLServer3"
--log=C:\Program Files\avs\var\avsql.log "SQLServer3\sql03inst2"
"(local)"
```

Backing up a database

To back up a specific database on a client, specify the client, instance, and database as the backup target between the quotation marks at the end of the command, using the format `client\instance/database`, or `(local)/database` if the database is in the local instance.

The following command performs a differential backup of the `DB1` database in the local instance on the `SQLServer1` client in the `SQL` domain, labels the backup `DB1Diff`, and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=differential --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log "(local)/DB1"
--label=DB1Diff
```

The following command performs a full backup of the DB3 database in the `sql01inst2` instance on the `SQLServer1` client in the SQL domain and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log
"SQLServer1\sql01inst2/DB3"
```

Backing up multiple databases

To back up multiple databases on a client, but not all databases, specify the instances as separate backup targets between quotation marks at the end of the command.

The following command performs a full backup of the DB3 database in the `sql01inst2` instance and the DB1 database in the local instance on the `SQLServer1` client in the SQL domain, and stores the backup on a Data Domain system:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log --ddr=true --ddr-index=2
"SQLServer1\sql01inst2/DB3" "(local)/DB1"
```

Backing up data in a failover cluster

To back up data in a failover cluster, run the CLI on the active node with the cluster client, and specify the cluster client as the client to back up. If you are backing up the local instance, then you must use the `--sqlserver` option and specify the virtual server for the SQL Server.

The following command performs a full backup of the DB1 database in the local instance on the `SQLCluster1` cluster client in the SQL domain, labels the backup `DB1Full`, and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLCluster1
--log=C:\Program Files\avs\var\avsql.log --sqlserver=SQLCluster1
"(local)/DB1" --label=DB1Full
```

The following command performs a transaction log backup of the DB3 database in the `sql01inst2` instance on the `SQLCluster1` cluster client in the SQL domain, and stores the backup on the Avamar server:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=incremental --path=/SQL/SQLCluster1
--log=C:\Program Files\avs\var\avsql.log --sqlserver=SQLCluster1
"SQLCluster1\sql01inst2/DB3"
```

Backing up data in an AlwaysOn availability group

To back up data in an AlwaysOn availability group:

- ◆ Run the CLI on the primary replica.
- ◆ Specify the cluster client for the availability group listener as the client to back up.
- ◆ Specify the availability group listener with the `--sqlserver` and `--hostname` options.
- ◆ Specify the availability group listener, the availability group, and the database in quotation marks using the format `"listener\group/database"`. To back up all databases in an availability group, specify `"listener\group"`.

The following command performs a full backup of all databases on the primary replica using the `SQLGroup2` cluster client for the `SQLGroup2` availability group and the `SQLGroupListener` availability group listener. The `SQLGroup2` cluster client is in the SQL domain on the Avamar server, and the backup is stored on the second Data Domain system that was added to the Avamar configuration:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLGroupListener
--log=C:\Program Files\avs\var\avsql.log
--backup-preference=primary --hostname=SQLGroupListener
--sqlserver=SQLGroupListener --ddr=true --ddr-index=2
"SQLGroupListener\SQLGroup2"
```

Backing up to a Data Domain system

To store a backup on a Data Domain system instead of on the Avamar server, use the `--ddr` and `--ddr-index` options.

The following command performs an incremental backup of all SQL Server data on the `SQLServer2` client in the SQL domain and stores the backup on the first Data Domain system that you added to the Avamar configuration:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=MCUser1 --brtype=incremental --path=/SQL/SQLServer2
--log=C:\Program Files\avs\var\avsql.log --ddr=true --ddr-index=1 "
```

Excluding data from a backup

To exclude an instance or database from a backup of an entire client, use the `--exclude` option. If you exclude an instance, you can include an individual database from the instance in the backup using the `--include` option.

The following command performs a full backup of all SQL Server data except for the local instance on the `SQLServer1` client in the SQL domain, and stores the backup on the Avamar server with an IP address of 12.34.56.78:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log --exclude=(local) " "
```

The following command performs the same backup as the previous example, but includes the database named DB3 in the backup:

```
avsql --operation=backup --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log --exclude=(local)
--include=(local)/DB3 " "
```

Example restore commands

The following topics provide example restore commands.

Restoring to the original location

To restore a backup to its original location, run the CLI on the original client, use the `--path` option to specify the domain and client, and use the `--labelnum` or `--label` option to specify the backup to restore.

Restoring an entire backup to the original location

The following command performs a tail-log backup and restores the backup labeled 223 to its original location of the `SQLServer3` client in the `SQL` domain:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --logtail --labelnum=223
--usesqlreplaceoption --log=C:\Program Files\avs\var\avsql.log
```

Restoring the local instance to the original location

The following command performs a tail-log backup and restores the local instance from the backup labeled `Backup6` to its original location on the `SQLServer3` client in the `SQL` domain:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --logtail --label=Backup6
(local) --log=C:\Program Files\avs\var\avsql.log
```

Restoring a database to the original location

The following command performs a tail-log backup and restores the `DB3` database from the `Instance2` instance in the backup labeled 842 to its original location on the `SQLServer3` client in the `SQL` domain:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --logtail
--usesqlreplaceoption --labelnum=842 Instance2/DB3
--log=C:\Program Files\avs\var\avsql.log
```


Restoring a filegroup or file to the original location

The following command performs a tail-log backup and restores the following items in the backup labeled 842 to the original location on the `SQLServer3` client in the `SQL` domain:

- ◆ `FG1` filegroup for the `DB1` database in the local instance
- ◆ `File1` file in the `FG2` filegroup for the `DB2` database in the local instance

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --logtail --labelnum=842
--restore-filegroup-list=(local)/DB1/FG1 (local)/DB2/DB2_FG2_File1
--log=C:\Program Files\avs\var\avsql.log
```

Restoring to a new database in the original instance

To restore a backup to its original location but with a new name, run the CLI on the original client, specify the original database name at the end of the command, and use the `--newdbname` option to specify the new database name.

The following command performs a tail-log backup and then restores the `DB2` database in the local instance from the backup labeled `DB2Full1` to its original location of the `SQLServer2` client in the `SQL` domain, but with a new name of `DB6`:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer2 --logtail --newdbname=DB6
--log=C:\Program Files\avs\var\avsql.log --label=DB2Full1
(local)/DB2
```

Restoring to a different instance

To restore a backup to a different instance, run the CLI on the destination client, use the `--path` option to specify the original domain and client, use the `--target` option to specify the new client and instance, and set `--restore-destination=multiple`.

Restoring a database to a different instance on the same client

The following command restores the `DB3` database from `Instance2` in the backup labeled 239 to an instance named `Instance3` on the `SQLServer1` client:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer1 --target=SQLServer1\Instance3
--restore-destination=multiple --labelnum=239 Instance2/DB3
--log=C:\Program Files\avs\var\avsql.log
```

Restoring multiple databases to a different instance on a different client

The following command restores the `DB3` database from `Instance2` and `DB1` from `Instance1` on the `SQLServer1` client in the `SQL` domain in the backup labeled `SQL1Backup8` to `Instance3` on the `SQLServer2` client:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer1 --target=SQLServer2\Instance3
--label=SQL1Backup8 Instance2/DB3 Instance1/DB1
--log=C:\Program Files\avs\var\avsql.log
```

You must run the previous command from a command line on the `SQLServer2` (destination) client.

Restoring files to a different location on a different client

The following command restores the File1 and File2 files in C:\SQLData from the (local)/DB3 database on the SQLServer1 client to C:\SQL on the SQLServer2 client in the Instance2/DB8 database:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer1 --target=SQLServer2\Instance2
--newdbname=DB8 --redirect="C:\SQLData\File1"="C:\SQL\File1",
"C:\SQLData\File2"="C:\SQL\File2" --labelnum=163 (local)/DB3
--log=C:\Program Files\avs\var\avsql.log
```

Restoring to a file

To restore a SQL Server database from an Avamar backup to operating system files, use the `--redirecttofile=true` option and specify the target path for the files using the `--target` option.

The following command restores the DB4 database in the local instance from the backup with a label number of 423 to the temp directory on the C:\ drive of the original location of the SQLServer1 client in the SQL domain:

```
avsql --operation=restore --id=AvamarAdmin --ap=password
--server=12.34.56.78 --logtail=false --path=/SQL/SQLServer1
--labelnum=423 --redirecttofile=true --target=C:\temp (local)/DB4
--log=C:\Program Files\avs\var\avsql.log
```

You must run the previous command from a command line on the original client.

The following command restores the DB2 database in Instance3 from the backup labeled Backup22 to the temp directory on the C:\ drive of the WinServer3 client in the ITServers domain, when WinServer3 is not the original client on which the backup was performed:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --label=Backup22 --target=C:\temp
--redirecttofile=true --path=/ITServers/WinServer3 Instance3/DB2
--log=C:\Program Files\avs\var\avsql.log
```

You must run the previous command from a command line on the WinServer3 (destination) client.

Restoring system databases

To restore SQL Server system databases from an Avamar backup, run the CLI on the original client and use the following options:

- ◆ `--handle-sql-services-automatically`
- ◆ `--restoresystem`
- ◆ `--use-sql-replace-option`

In addition, do *not* use the `--logtail` option.

The following command restores the backup labeled 395 to its original location of the SQLServer3 client in the SQL domain, including the system databases in the backup:

```
avsq1 --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLServer3 --restoresystem
--handle-sql-services-automatically --logtail=false
--use-sql-replace-option --labelnum=395
--log=C:\Program Files\avs\var\avsq1.log
```

Restoring in a failover cluster

To restore data in a failover cluster, run the CLI on the active node. If you are backing up the local instance, then you must use the `--sqlserver` option and specify the virtual server for the SQL Server.

The following command restores the DB1 database to its original location in the local instance on the SQLCluster1 cluster client in the SQL domain:

```
avsq1 --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --brtype=full --path=/SQL/SQLCluster1
--sqlserver=SQLCluster1 "(local)/DB1" --label=DB1FULL
--log=C:\Program Files\avs\var\avsq1.log
```

Restoring to an AlwaysOn availability group

To restore data to an AlwaysOn availability group, run the CLI on the primary replica. The options you use in the command depend on the restore destination. The following topics provide details:

- ◆ [“Restoring to the original availability group” on page 187](#)
- ◆ [“Restoring to a new database in the original availability group” on page 188](#)
- ◆ [“Restoring to a different availability group” on page 188](#)

The steps to restore to a different instance or to operating system files are the same whether you restore from a backup of databases on a stand-alone server, in a cluster, or in an availability group. [“Restoring to a different instance” on page 185](#) and [“Restoring to a file” on page 186](#) provide details.

Restoring to the original availability group

To restore a backup to its original location in an availability group:

- ◆ Use the `--path` option to specify the Avamar domain and the cluster client for the availability group listener.
- ◆ Use the `--labelnum` or `--label` option to specify the backup to restore.
- ◆ Specify the availability group listener in the `--sqlserver` and `--hostname` options.
- ◆ Use the `--recovery-primary-only` option to control whether to restore to only the primary replica or to both the primary and all secondary replicas.

The following command performs a tail-log backup and restores the DB3 database in the backup labeled 842 to its original location on all replicas in the SQLGroup3 availability group client in the SQL domain:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLGroup3 --sqlserver=SQLGroupListener
--hostname=SQLGroupListener --logtail --usesqlreplaceoption
--log=C:\Program Files\avs\var\avsql.log
--recovery-primary-only=false --labelnum=842
SQLGroupListener\SQLGroup3/DB3
```

The following command performs the same restore as the previous command, but restores to only the primary replica:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLGroup3 --sqlserver=SQLGroupListener
--hostname=SQLGroupListener --logtail --usesqlreplaceoption
--log=C:\Program Files\avs\var\avsql.log
--recovery-primary-only=true --labelnum=842
SQLGroupListener\SQLGroup3/DB3
```

When you restore the databases on only the primary replica, the corresponding database on the secondary replicas is in a restoring state. To restore the databases on the secondary replicas as part of the availability group, manually prepare and restore the databases, and join them to the availability group on the secondary replica.

Restoring to a new database in the original availability group

To restore a backup to its original location but with a new name, specify the original database name at the end of the command, and use the `--newdbname` option to specify the new database name.

The following command performs a tail-log backup and then restores the DB2 database in the SQLGroup2 availability group from the backup labeled DB2Full to its original location, but with a new name of DB6:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLGroup2 --sqlserver=SQLGroupListener
--hostname=SQLGroup2Listener --logtail --newdbname=DB6
--log=C:\Program Files\avs\var\avsql.log --label=DB2Full
SQLGroupListener\SQLGroup2/DB2
```

Restoring to a different availability group

The following command restores the DB3 database from the SQLGroup2 availability group in the backup labeled 239 to an availability group named SQLGroup3:

```
avsql --operation=restore --server=12.34.56.78 --id=AvamarAdmin
--ap=password --path=/SQL/SQLGroup3 --sqlserver=SQLGroupListener
--hostname=SQLGroupListener --target=SQLGroup3Listener\SQLGroup3
--restore-destination=multiple --labelnum=239
SQLGroupListener\SQLGroup2/DB3
--log=C:\Program Files\avs\var\avsql.log
```

Run the previous command on the primary replica of the target cluster with the SQLGroup3 availability group.

Point-in-time restore

To restore to a specific point in time, set `--pitormark=pointintime` and specify the date and time with the `--pitormarkstr` option.

To restore to a named mark, set `--pitormark=markname`, specify the named mark using the `--pitormarkstr` option, and specify whether to include the mark in the recovery using the `--recoverypoint` option. If named marks are not unique in the transaction log, then use the `--aftertime` option to specify a date and time to locate the mark.

The following command restores the DB3 database in the local instance from the backup labeled `FullBackup2` to its original location and a point in time of January 15, 2013 at 2:15:45 p.m.:

```
avsql --operation=restore --id=AvamarAdmin --ap=password
--server=12.34.56.78 --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log --pitormark=pointintime
--pitormarkstr=2013-01-15T14:15:45 --label=FullBackup2 (local)/DB3
```

The following command restores the DB3 backup in `Instance2` from the backup labeled `452` to its original location and a named mark of `M2`, and the recovery includes the transaction at the named mark:

```
avsql --operation=restore --id=AvamarAdmin --ap=password
--server=12.34.56.78 --path=/SQL/SQLServer1
--log=C:\Program Files\avs\var\avsql.log --pitormark=markname
--pitormarkstr=M2 --recoverypoint=atmark --labelnum=452
Instance2/DB3
```

Viewing backup metadata

To view the metadata for a backup as screen output, use the `--metadata` option with the `restore` operation.

The following command displays metadata for the backup with a label number of `22`:

```
avsql --operation=restore --id=AvamarAdmin --ap=password
--server=12.34.56.78 --path=/SQL/SQLServer1 --metadata
--labelnum=22
```


GLOSSARY

This glossary provides definitions of terms used in this guide.

A

activation	See “ client activation ”.
administrator	Person who normally installs, configures, and maintains software on network computers, and who adds users and defines user privileges.
alias	A SQL Server feature that enables you to configure a name on a computer with SQL Server that points to a server, instance, or database on either the local computer or on a different computer.
AlwaysOn availability group	In a cluster environment with SQL Server 2012, a group of databases stored locally on each node that can fail over from one node to another for high availability.
authentication	The method by which users and applications, such as Avamar client software, connect to SQL Server. Authentication methods include <i>Windows authentication</i> and <i>SQL Server authentication</i> .
Avamar Administrator	A graphical management console software application that is used to remotely administer an Avamar system from a supported Windows or client computer.
Avamar client	A computer or workstation that runs Avamar software and accesses the Avamar server over a network connection. Avamar client software comprises a <i>client agent</i> and one or more <i>plug-ins</i> .
Avamar server	The server component of the Avamar client/server system. Avamar server is a fault-tolerant, high-availability system that efficiently stores the backups from all protected clients. It also provides essential processes and services required for data restores, client access, and remote system administration. Avamar server runs as a distributed application across multiple networked storage nodes.
avagent	The name of the <i>client agent</i> process.
availability group	See “ AlwaysOn availability group ”.
availability group listener	A virtual network name (VNN) to which clients can connect to access a database in an availability replica of an availability group.
availability replica	A SQL Server instance on a cluster node that hosts an <i>AlwaysOn availability group</i> .
avsql	The application executable and command line interface for the Avamar Plug-in for SQL Server. You can use <code>avsql</code> to perform on-demand backups and restores of SQL Server data.
avtar	The Avamar process that performs backups and restores.

B

- backup** A point-in-time copy of client data that can be restored as individual files, selected data, or as an entire backup.
- bare metal recovery (BMR)** The process of restoring data from a backup to recover a computer system from “bare metal,” or without previously installed software or operating system.
- browse** The process of viewing data that is available for backup on a client computer or restore from the Avamar server.
- bulk-logged recovery model** A type of SQL Server *recovery model* that permits high-performance bulk copy operations and reduces log space usage by using minimal logging for most bulk operations. This recovery model enables you to recover to the end of any backup. *Point-in-time restore* is not supported.

C

- client activation** The process of passing the client ID (CID) back to the client, where it is stored in an encrypted file on the client file system.
- client agent** A platform-specific software process that runs on the client and communicates with the MCS and with any plug-ins installed on that client.
- client registration** The process of establishing an identity with the Avamar server. When Avamar recognizes the client, it assigns a unique client ID (CID), which it passes back to the client during activation.
- cluster** Two or more independent network servers, usually with exact copies of the same software and data, that operate and appear to clients as if they are a single unit. A cluster configuration enables work to be shifted from one server to another, providing high availability, which allows application services to continue despite most hardware or software failures.
- Cluster Configuration Tool** Avamar configuration wizard to configure the cluster client.

D

- Data Domain system** Disk-based deduplication appliances and gateways that provide data protection and disaster recovery (DR) in the enterprise environment.
- database** A collection of data arranged for ease and speed of update, search, and retrieval by computer software.
- database mirroring** A high availability strategy where SQL Server maintains two copies of a single database that must reside on different SQL Server instances.
- dataset** A policy that defines a set of files, directories, and file systems for each supported platform that are included or excluded in backups across a group of clients. A dataset is a persistent and reusable Avamar policy that can be named and attached to multiple groups.
- default instance** See “[local instance](#)”.

differential backup	A type of backup that includes only data that changed after the last <i>full backup</i> . The backup also copies relevant portions of the transaction logs.
disaster recovery	Recovery from any disruptive situation, such as hardware failure or software corruption, in which ordinary data recovery procedures are not sufficient to restore a system and its data to normal day-to-day operations. A disaster recovery may or may not be a <i>bare metal recovery</i> .
DNS	Domain Name Server. A dynamic and distributed directory service for assigning domain names to specific IP addresses.
domain	A feature in Avamar Administrator that is used to organize large numbers of clients into named areas of control and management.

F

Failover Cluster Instances (FCIs)	A single <i>instance</i> of SQL Server that is installed across Windows Server Failover Clustering (WSFC) nodes. On the network, an FCI appears to be an instance of SQL Server running on a single computer, but the FCI provides failover from one WSFC node to another if the current node becomes unavailable.
file system	Software interface used to save, retrieve, and manage files on storage media by providing directory structures, data transfer methods, and file association.
filegroup	Named collections of SQL Server files that are used to help with data placement and administrative tasks such as backup and restore operations. The two types of filegroups are the <i>primary filegroup</i> and <i>user-defined filegroups</i> .
full backup	A type of backup that includes the entire database, including all objects, system tables, and data, as well as the transaction log.
full recovery model	A type of SQL Server <i>recovery model</i> that logs all database transactions in the transaction log. This type of recovery model enables you to perform a point-in-time restore.

G

group	A level of organization in Avamar Administrator for one or more Avamar clients. All clients in an Avamar group use the same group policies, which include the <i>dataset</i> , <i>schedule</i> , and <i>retention policy</i> .
group policy	The <i>dataset</i> , <i>schedule</i> , and <i>retention policy</i> for all clients in an Avamar group.

I

incremental backup	A type of backup that includes only the transaction log. Also called a <i>transaction log backup</i> .
instance	A single occurrence of the SQL Server Database Engine. You can install multiple instances on a single computer. The two types of instances are the <i>local instance</i> , also called the <i>default instance</i> , and <i>named instances</i> .

L

- LAN** Local Area Network.
- local instance** The default *instance* on a computer with SQL Server. The name of the local instance is the name of the local computer.

M

- Management Console Server (MCS)** The component that provides centralized administration (scheduling, monitoring, and management) for the Avamar server. The MCS also runs the server-side processes used by *Avamar Administrator*.
- mark** See “[named mark](#)”.
- master database** The *system database* that records all of the system level information for a SQL Server system.
- metadata** Data about the backup, including information about the original database files, the backup types, the date and time of the backup, and other information necessary for restore.
- mirroring** See “[database mirroring](#)”.
- model database** The *system database* that provides the template for all databases that you create on a SQL Server *instance*.
- msdb database** The *system database* for scheduling alerts and jobs, and for recording operators. The msdb database also contains history tables, such as the backup and restore history tables.
- multi-streaming** The process of backing up or restoring SQL Server data using multiple parallel data streams. Multi-streaming enables you to improve backup and restore performance in most environments.

N

- named instance** All instances of the database engine other than the *local instance*. You specify the name of the instance when you install the instance.
- named mark** The user-defined name for a starting point of a specific type of transaction in the transaction log. You can perform a *point-in-time restore* for a database to a named mark.

P

- plug-in** An Avamar software process that recognizes a particular kind of data resident on that client.
- plug-in options** Options that you specify during backup or restore to control backup or restore functionality.
- point-in-time restore** The process of restoring a database with the full recovery model to a specific date and time or to a *named mark* in the transaction log.

policy	A set of rules for client backups that can be named and applied to multiple groups. Groups have dataset, schedule, and retention policies.
primary data file	The data file that is the starting point of the database and points to the other files in the database. Every database has one primary data file. The recommended filename extension for primary data files is <code>.mdf</code> .
primary filegroup	The <i>filegroup</i> that contains the primary data file and any other files not specifically assigned to another filegroup. All pages for the system tables are allocated in the primary filegroup.
primary replica	The <i>availability replica</i> that handles read/write activity from clients and sends transaction log updates to the secondary replicas.
R	
recovery model	A database property that controls how transactions are logged, whether the transaction log requires and allows backups, and the available restore operations. Three recovery models exist: simple, full, and bulk-logged.
recovery operation	The type of recovery to perform when you restore SQL Server data. The recovery operation controls the status of restored databases after the restore. The three recovery models are RECOVERY, NORECOVERY, and STANDBY.
redirected restore	The process of restoring a backup to a different location than the original location where the backup occurred.
registration	See “client registration” .
replica	See “availability replica” .
Resource database	A read-only <i>system database</i> that contains copies of all system objects that ship with SQL Server. The Avamar Plug-in for SQL Server cannot back up the Resource database because SQL Server cannot back up the Resource database.
restore	An operation that retrieves one or more file systems, directories, files, or data objects from a backup and writes the data to a designated location.
retention	The time setting to automatically delete backups on an Avamar server. Retention can be set to permanent for backups that should not be deleted from an Avamar server. Retention is a persistent and reusable Avamar policy that can be named and attached to multiple groups.
S	
schedule	The ability to control the frequency and the start and end time each day for backups of clients in a group. A schedule is a persistent and reusable Avamar policy that can be named and attached to multiple groups.
secondary data files	The data files other than the <i>primary data file</i> for a database. Some databases may not have any secondary data files, while others have several secondary data files. The recommended filename extension for secondary data files is <code>.ndf</code> .

secondary replica	An <i>availability replica</i> that contains a copy of the databases for the availability group. The database copies synchronize with the databases on the primary replica through transaction log updates from the primary replica.
simple recovery model	A type of SQL Server <i>recovery model</i> that permits high-performance bulk copy operations and reclaims log space to minimize space requirements. This recovery model enables you to recover to the end of any backup. Databases with the simple recovery model do not support transaction log backups or point-in-time restore.
SQL Server authentication	An <i>authentication</i> method by which users and applications, such as Avamar client software, connect to SQL Server using a login account defined in SQL Server.
SQL Server Management Studio (SSMS)	A software application with script editors and graphical tools to configure, manage, and administer SQL Server components.
sqlcmd utility	A command line utility for entering <i>Transact-SQL</i> statements and scripts.
system databases	Databases that store system-level information, objects, and templates for a SQL Server <i>instance</i> . The master, msdb, model, and Resource databases are system databases.
T	
tail-log backup	A transaction log backup of log records that have not yet been backed up (the “tail of the log”). A tail-log backup occurs before you restore data from a backup. The backup prevents work loss and keeps the log chain intact.
Transact-SQL	An extension of Structured Query Language (SQL). You can use Transact-SQL statements to back up and restore SQL Server data.
transaction log	A log file that contains a history of database transactions.
transaction log backup	A type of backup that includes only the transaction log. Also called an <i>incremental backup</i> .
U	
user-defined filegroup	Any <i>filegroup</i> that you define in SQL Server. Log files are never part of a filegroup.
V	
virtual device interface (VDI)	The Microsoft application programming interface (API) that the Avamar Plug-in for SQL Server uses to back up and restore SQL Server data.
virtual machine (VM)	A computer that is a software implementation of a computer. Virtual machines are used to run different operating systems at the same time on one physical computer. Each operating system runs in its own isolated execution environment.
W	
Windows authentication	An <i>authentication</i> method by which users and applications, such as Avamar client software, connect to SQL Server using a Windows user account that has privileges in SQL Server.