



EMC[®] Avamar[®] 6.1 for SQL Server

User Guide

P/N 300-013-357
REV 07

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

Published August, 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	14
Avamar components	14
How Avamar connects to SQL Server	15
How Avamar works in a high availability environment	15
Using Avamar with Data Domain	20
Server virtualization	21
Backup	21
Backup types	21
Choosing a backup type based on the recovery model	22
Mixed backups.....	23
Canceling a backup.....	24
Multi-streaming.....	24
Transparent Data Encryption	24
Restore	25
Restoring to the original location.....	25
Restoring to a new database in the original instance	26
Restoring to a different instance on the original server	26
Restoring to an instance on a different server	26
Restoring to a file	27
Restoring a database in an AlwaysOn availability group	27
Restoring a database with an intact log file	27
Restoring system databases.....	28
Tail-log backup	29
Point-in-time restore	29
SQL Server recovery operations	30
Disaster recovery	30
New features in this release	31
New features in release 6.1 SP1	31
New features in release 6.1	31
SQL Server 2012 support and requirements	32
Known limitations	33
Avamar plug-in and server version compatibility issues.....	33
Backing up databases with FILESTREAM data is not supported	33
Different database sizes appear in Avamar Administrator.....	34
Viewing partial results after a browse timeout is not supported	34
Backing up the Resource database is not supported	34
Error appears for a forced incremental backup after a restore	34

Chapter 2

Installation

System requirements	36
Supported configurations.....	36
Hardware requirements.....	37
SQL Server requirements.....	37
Security requirements	39
Avamar server dependencies	39

Installation road map	40
Standalone installation road map	40
Cluster installation road map	40
Installing the Avamar Client for Windows.....	41
Installing the Avamar Plug-in for SQL Server	43
Registering the client	44
Installing the Avamar Cluster Client	45
Verifying the configuration	47
Upgrade	48
Upgrading the Avamar Plug-in for SQL Server	48
Upgrading to Avamar Cluster Client	48
Uninstall	50
Uninstalling the Avamar Cluster Client.....	50
Uninstalling the Avamar Plug-in for SQL Server.....	51
Uninstalling the Avamar Client for Windows	51

Chapter 3 Backup

On-demand backup	54
Scheduled backups.....	61
Creating a dataset	61
Creating a group.....	71
Enabling scheduled backups	75
Multi-streaming	75
When to use multi-streaming.....	76
Calculating the number of streams for a database.....	76
Understanding the order of databases in a backup	77
Specifying the maximum number of streams	78
Specifying the minimum stream size	79

Chapter 4 Restore

Restore requirements.....	86
Software requirements for restore	86
Tail-log backup and point-in-time restore requirements.....	86
Requirements to restore a database in an AlwaysOn availability group..	87
Requirements to restore secondary database files	87
Requirements to restore the Report Server database	87
SQL Server write permission requirements	87
Finding a backup.....	88
When to find a backup by date	88
How to find a backup by date	88
When to find a backup by file or folder	91
How to find a backup by file or folder	91
Restoring to the original location.....	94
Understanding the options when restoring to the original location	94
How to restore to the original location	95
Restoring to a new database in the original instance	98
Restoring to a different instance.....	101
Restoring to a file	107
Restoring to a file with the SQL Server plug-in	107
Restoring to a file with the Windows file system plug-in.....	112
Restoring a database with SQL Server tools.....	115
Restoring system databases.....	121
Restoring system databases automatically.....	121

	Restoring system databases manually	124
	Restoring a database in an AlwaysOn availability group	127
	Restoring a database on the primary replica.....	127
	Restoring a database on a secondary replica.....	128
	Restoring a database with an intact log file	128
	Setting restore options.....	129
	General restore options.....	130
	Recovery operation options.....	131
	System database restore options	132
	Redirected restore options	133
	Authentication options.....	134
	Point-in-time recovery options.....	134
Chapter 5	Disaster Recovery	
	Preparing for disaster recovery	138
	Performing disaster recovery	138
Appendix A	Plug-in Options	
	How to set plug-in options	140
	Backup options.....	140
	Restore options.....	143
Appendix B	Command-Line Interface	
	Understanding the SQL Server plug-in CLI	148
	CLI architecture	148
	Launching the CLI.....	148
	Available operations	148
	Options.....	149
	Using the CLI in a cluster	149
	Password encoding.....	150
	Help	150
	Command reference	151
	Synopsis	151
	Operations.....	151
	Options.....	152
	CLI examples	163
	Example browse commands.....	163
	Example backup commands.....	163
	Example restore commands	166

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 representative if a product does not function properly or does not function as described in this document.

NOTICE

This document was accurate at publication time. New versions of this document might be released on EMC Online Support. Check EMC Online Support 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:

- ◆ 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
07	August 31, 2013	Added requirements to “Software requirements for restore” on page 86 .
06	June 17, 2013	Removed step for configuring the cluster client from “AlwaysOn availability groups” on page 17 . Updated “AlwaysOn availability group requirements” on page 38 to include the following information: <ul style="list-style-type: none"> • How to determine the node that is hosting the primary replica. • How to configure scheduled backups of all replicas.
05	March 1, 2013	Clarified support for SQL Server 2012 and Windows Server 2012 in “New features in this release” on page 31 .
04	December 7, 2012	Added requirements for SQL Server Management Tools and the SQL Client Connectivity SDK in “SQL Server feature requirements” on page 37 .
03	October 25, 2012	Updates for release 6.1 Service Pack 1: <ul style="list-style-type: none"> • Added support for SQL Server 2008, SQL Server 2008 R2, and SQL Server 2012 on Windows Server 2012. • Added support for the --label option for both backup and restore using the command-line interface. • Added details on the --restore-destination option for restores using the command-line interface. • Added details on the --encrypt and --encrypt-strength options for backup and restore using the command-line interface.
02	July 31, 2012	Updated “Where to get help” on page 10 in the Preface.
A01	April 25, 2012	First release of Avamar 6.1.

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 Data Domain 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:

Normal	Used in running (nonprocedural) text for: <ul style="list-style-type: none"> Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, and utilities URLs, pathnames, file names, directory names, computer names, links, groups, service keys, file systems, and notifications
Bold	Used in running (nonprocedural) text for names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, and man pages Used in procedures for: <ul style="list-style-type: none"> Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus What the user specifically selects, clicks, presses, or types
<i>Italic</i>	Used in all text (including procedures) for: <ul style="list-style-type: none"> Full titles of publications referenced in text Emphasis, for example, a new term Variables
Courier	Used for: <ul style="list-style-type: none"> System output, such as an error message or script URLs, complete paths, file names, prompts, and syntax when shown outside of running text
Courier bold	Used for specific user input, such as commands
<i>Courier italic</i>	Used in procedures for: <ul style="list-style-type: none"> Variables on the command line User input variables
< >	Angle brackets enclose parameter or variable values supplied by the user
[]	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 Service.

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.

Live chat

To engage EMC Customer Service 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 Service, 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 Service.

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 2005, 2008, or 2012 environment:

◆ Architecture	14
◆ Backup	21
◆ Restore	25
◆ Disaster recovery.....	30
◆ New features in this release	31
◆ Known limitations	33

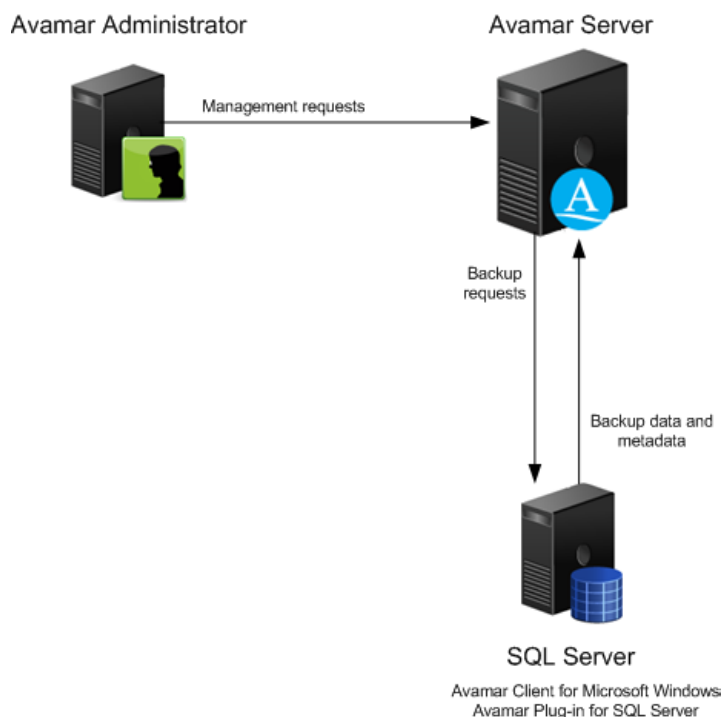
Architecture

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

- ◆ “Avamar components” on page 14
- ◆ “How Avamar connects to SQL Server” on page 15
- ◆ “How Avamar works in a high availability environment” on page 15
- ◆ “Using Avamar with Data Domain” on page 20
- ◆ “Server virtualization” on page 21

Avamar components

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



You must install both the Avamar Client for Windows and the Avamar Plug-in for SQL Server on the SQL server:

- ◆ **Avamar Client for Windows** — When you install the Avamar Client for Windows, the installation includes the the Avamar Plug-in for Windows and the Avamar agent, which is required for the Avamar Plug-in for SQL Server. You can use the Avamar Plug-in for Windows 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 that is used to remotely administer 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.

Avamar creates one or more VDI devices, and then executes the necessary SQL commands to back up or restore the database to or from the VDI devices. The Avamar **avtar** program is then spawned in a standard input/output (stdio) mode. For backups, data is read from the VDI device and written to **avtar** standard input, where **avtar** reads it and sends the data to either the Avamar server or a configured Data Domain system. For restores, data is read from standard output and written to the VDI device, which communicates with SQL server to restore the databases.

The number of VDI devices that Avamar 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 24](#) provides details on streams.

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 using a Windows user account that has privileges in SQL Server. The account should be a domain account with the proper administrative privileges.
- ◆ With SQL Server authentication, Avamar connects to SQL Server 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.

The following topics provide details on how Avamar works in each strategy:

- ◆ [“Failover Cluster Instances” on page 15](#)
- ◆ [“AlwaysOn availability groups” on page 17](#)
- ◆ [“Database mirroring” on page 19](#)

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:

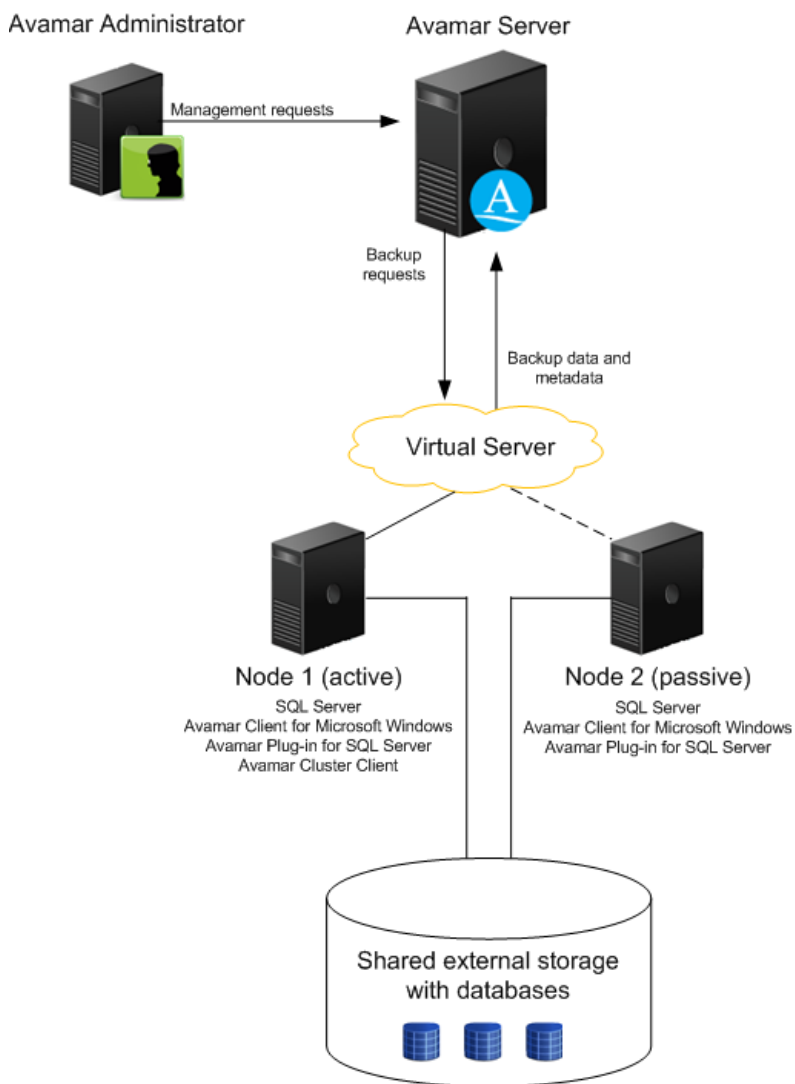
- ◆ In an active/passive configuration, SQL Server runs on the active node, which is the primary node. The passive node is a standby node. SQL Server does not run on the passive node. If a failure occurs on the primary active node, then SQL Server fails over to the passive node. The passive node becomes 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.
- ◆ 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 while SQL Server is running on either the primary active node or on a standby node. To enable Avamar to back up and restore SQL Server data in a cluster:

1. Install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on the same nodes in the cluster on which you install SQL Server.
2. Register each node in the cluster as a client with the Avamar server.
3. Use the Cluster Configuration Tool to install the Avamar Cluster Client on the active node in an active/passive configuration or on both active nodes in an active/active configuration. The Cluster Configuration Tool also registers the virtual server with the Avamar server so that you can perform backups and restores regardless of which node is hosting SQL Server. The Avamar Cluster Client manages Avamar functionality in a cluster.
4. When you perform a backup or restore, select the virtual server for SQL Server as the client to back up or the target client for the restore.

Avamar backs up or restores the SQL Server data regardless of which cluster node is active.

The following figure illustrates an active/passive cluster environment with SQL Server and Avamar. Avamar communicates with the SQL Server virtual server, which is currently active on Node 1, to back up and restore the SQL Server databases on the shared external storage in the cluster.



AlwaysOn availability groups

SQL Server 2012 introduces high availability of groups of databases through 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 a set of user databases together in an availability group. Databases in an availability group must use the full recovery model. The databases fail over together from one node to another as part of the availability group.

A SQL Server instance on a cluster node that hosts an availability group is called an *availability replica*. Each availability replica of a given 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 given 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. It 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, becoming a secondary replica.

Avamar can perform both on-demand and scheduled backups of databases in only the primary replica for an availability group. As a result, you should perform a full backup of the databases in an availability group after it fails over from one node to another.

You can use the backup from the primary replica to restore the databases in either the primary replica or the secondary replicas, if necessary.

To use Avamar to back up and restore databases in availability groups in a cluster:

1. Install the Avamar Client for Windows and the Avamar Plug-in for SQL Server on the same nodes in the cluster on which you install SQL Server.
2. Register each node in the cluster as a client with the Avamar server.
3. When you perform a backup of a database in an availability group, select the hostname of the physical node that is hosting the primary replica as the client to back up.
4. To restore a database in the primary replica:
 - a. Remove the database from the availability group.
 - b. Select the hostname of the physical node that is hosting the primary replica as the Avamar client from which to restore, and then use the Avamar Plug-in for SQL Server to restore the database to its original location, as described in [“Restoring to the original location”](#) on page 94.
 - c. Add the database back to the availability group.

NOTICE

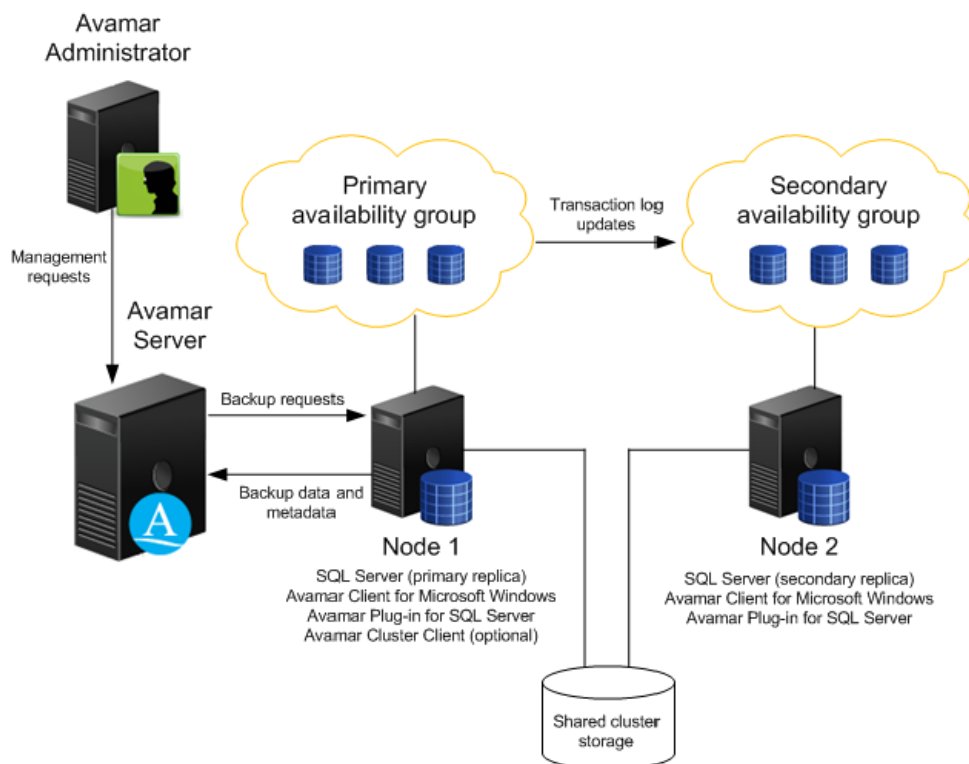
After you remove the database from the availability group on the primary replica, the corresponding database on the secondary replicas are in a restoring state. There are two ways to restore the databases on the secondary replicas as part of the availability group. You can either delete the databases on the secondary replicas and then automatically re-create and synchronize the databases when you add the database back to the availability group, or you can 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. The SQL Server 2012 documentation available on the [Microsoft website](#) provides details.

5. To restore a database in a secondary replica:
 - a. Remove the database from the availability group.
 - b. Select the hostname of the physical node that is hosting the primary replica as the Avamar client from which to restore, and then use the Avamar Plug-in for SQL Server to restore the database to the corresponding SQL Server instance on the

other cluster node, as described in [“Restoring to a different instance” on page 101](#). During the restore, select NORECOVERY for the recovery operation in the plug-in options.

- c. Join the database to the to the availability group as described in the SQL Server 2012 documentation on the Microsoft website.

The following figure illustrates how Avamar connects to the primary replica for backup and restore.



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.

Mirroring is implemented on a per-database basis and 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 SQL Server databases that are mirrored. However, several conditions apply, as discussed in [“Database mirroring requirements” on page 39](#).

Using Avamar with Data Domain

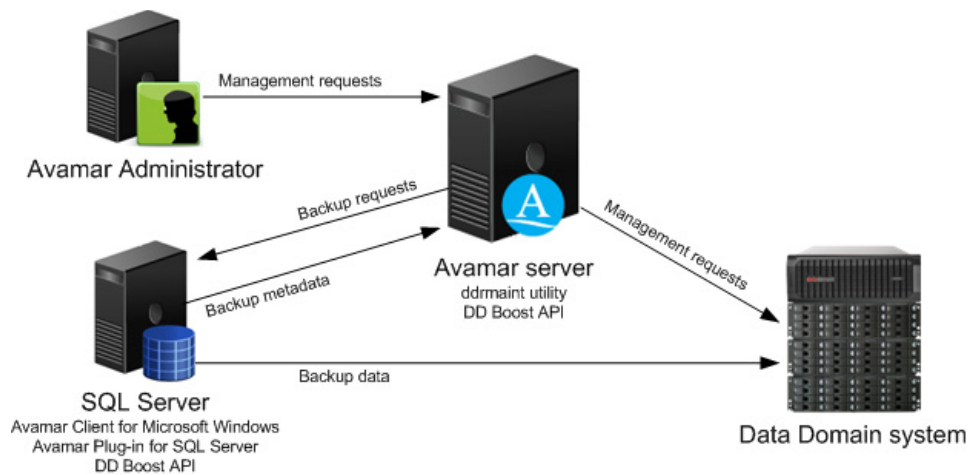
You can store SQL Server backups on either the Avamar server or on an EMC Data Domain[®] system. By default, backups are stored on the Avamar server.

Avamar clients use the DD Boost API to access a Data Domain system. The DD Boost API is installed automatically on the client computer when you install the Avamar client.

The **ddrmaint** utility implements all required operations on the Data Domain system for the Avamar server. The **ddrmaint** utility is installed on the utility node of a multi-node server, or the single node of a single-node server, during Avamar server installation. It is not installed on the data nodes of the Avamar server.

The **ddrmaint** utility uses the DD Boost API to connect to a Data Domain system. The DD Boost API is installed with the **ddrmaint** utility automatically when you install Avamar.

The following figure illustrates an environment where Avamar sends SQL Server backup data to a Data Domain system.



Additional details on supported environments and system requirements are available in the *EMC Avamar and Data Domain Integration Guide*.

To store backups on a Data Domain system, use Avamar Administrator to configure the Data Domain system for use, then select the Data Domain system during an on-demand backup or when configuring the dataset for a scheduled backup. The *EMC Avamar and Data Domain Integration Guide* provides details on how to configure the Data Domain system in Avamar Administrator. The individual backup procedures in this guide provide the steps on selecting the Data Domain system to use.

Keep in mind, however, that mixed backups are not supported. 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. [“Mixed backups” on page 23](#) provides additional details.

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 following guides provide details on additional system requirements, as well as installation and configuration procedures:

- ◆ *EMC Avamar for Hyper-V VSS User Guide*
- ◆ *EMC Avamar for VMware User Guide*

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 22](#) provides details.

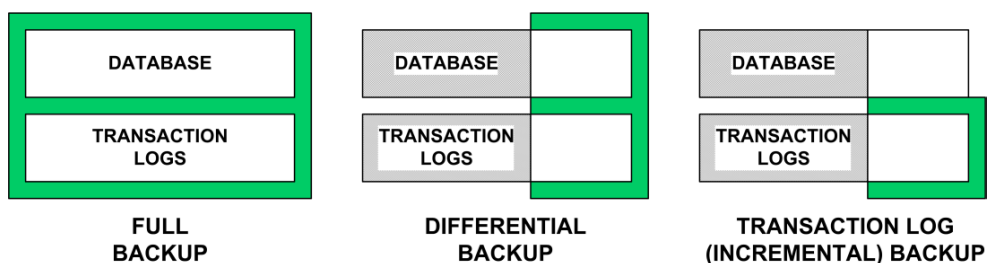
You can store backups on either the Avamar server or a Data Domain system. However, mixed backups are not supported. [“Mixed backups” on page 23](#) provides details.

You also can improve backup performance using multi-streaming, as discussed in [“Multi-streaming” on page 24](#).

Backup types

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

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



Full backups

Full backups always back up the entire database, including all objects, system tables, and data. As the backup operation progresses, it copies the transaction logs. This ensures that you can recover the complete database to the state it was in when the backup finished.

Differential backups

Differential backups back up any data that has changed since the last full backup. As the backup operation progresses, it copies relevant portions of the transaction logs. This ensures that you can recover the database to the state it was in when the backup finished. To restore the database, the Differential Restore restores the last full backup, followed by the differential backups performed after the full backup. Because it only saves changes to data, a differential backup is smaller and faster than a full backup, and can therefore be performed more often.

Differential backups are used primarily in heavily utilized systems where a failed database must be brought 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. They 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 that have taken place since 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 is truncated to remove any transactions that have been 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 one is found, the transaction log backup proceeds normally (that is, only backing up transaction logs). If a full backup is not found and the Force full backup option is selected (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. This is because 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 you include databases that use the simple recovery model in a transaction log backup with other databases that use other recovery models, then you can choose how Avamar handles the backups of the databases that use the simple recovery model. You can exclude the databases that use 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 SQL Server 2012 availability groups must use the full recovery model.

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

Mixed backups

You can store backups on either the Avamar server or a Data Domain system. However, mixed backups are not supported. 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 is if you perform a tail-log backup during a restore, then 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 are changing the backup storage target from one Data Domain system to another Data Domain system, and you want to be able to restore to a point in time between the two full backups, then 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, only a single backup set that includes both the new full backup and the forced incremental backup is created.

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, use the following steps:

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 107](#) provides details on restoring to a file, as well as examples of using SQL Server tools to restore from a file to SQL Server.

Canceling a backup

When you cancel a backup activity, always perform a new full backup immediately. This is because 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 backup to an operating system file, as described in [“Restoring to a file” on page 107](#), and then use SQL Server tools to restore the database to a certain point in time.

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 number of streams that you use for the backup is also used for the restore.

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

[“Multi-streaming” on page 75](#) provides details on how multi-streaming works, as well as how to set multi-streaming options to maximize performance in different environments.

Transparent Data Encryption

Avamar backups of SQL Server 2008, SQL Server 2008 R2, or SQL Server 2012 databases protected by Transparent Data Encryption (TDE) intentionally do not include the Database Encryption Key (DEK) in the backup.

Including the DEK defeats one of the primary reasons for using TDE, which is to encrypt data that is exported from the system, 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 (<http://technet.microsoft.com>), provides additional details.

Restore

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

- ◆ 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 database that is lost or corrupt but the database log file is intact.

There are several additional considerations when you restore system databases.

The following topics provide details on each of these restore scenarios:

- ◆ “Restoring to the original location” on page 25
- ◆ “Restoring to a new database in the original instance” on page 26
- ◆ “Restoring to a different instance on the original server” on page 26
- ◆ “Restoring to an instance on a different server” on page 26
- ◆ “Restoring to a file” on page 27
- ◆ “Restoring a database in an AlwaysOn availability group” on page 27
- ◆ “Restoring a database with an intact log file” on page 27
- ◆ “Restoring system databases” on page 28

When you perform a restore, Avamar provides several options to control restore behavior. The following topics provide an introduction to several key options:

- ◆ “Tail-log backup” on page 29
- ◆ “Point-in-time restore” on page 29
- ◆ “SQL Server recovery operations” on page 30

Restoring to the original location

When you use the Avamar Plug-in for SQL Server to restore a SQL Server instance, database, or filegroup to its original location, you can either perform a standard restore with a tail-log backup and recovery, or you can 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 have not been included 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 required, 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, which overrides a SQL Server safety check that is intended to prevent you from accidentally overwriting a different database or file. This safety check is described in the Microsoft Transact-SQL Reference Manual under the RESTORE command section.

Restoring 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.

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 included in the backup. [“Tail-log backup” on page 29](#) 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 29](#) provides additional details on point-in-time restore.

Restoring 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.

When you restore to a different instance on the original server, tail-log backup is not supported. 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 29](#) provides additional details on point-in-time restore.

Restoring 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.

When you restore to an instance on a different server, tail-log backup is not supported. 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 29](#) provides additional details on point-in-time restore.

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 are not provided by the Avamar Plug-in for SQL Server, then you can restore a SQL Server database from an Avamar backup 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\FILENAME
```

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.
- ◆ FILENAME is the name of the file. There may be multiple files for a single backup, depending on the number of streams that were used to perform 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.

Restoring a database in an AlwaysOn availability group

Because you can only back up an availability database on the primary replica, the steps to restore a database depend on whether the database is in an availability group on the primary replica or a secondary replica. [“Restoring a database in an AlwaysOn availability group” on page 127](#) 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 that occurred since 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 128](#) provides detailed steps.

Restoring system databases

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

It is more likely that you will need to restore system databases at the same time that you restore user databases. When you select both system and user databases for restore, the system databases are restored 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. If necessary, however, you can restore individual system databases and manually manage the services.

Restoring system databases automatically

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, the instance is automatically restarted.
- ◆ 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 121](#) provides detailed steps.

Restoring system databases manually

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 124](#) provides detailed steps.

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 have not been backed up. The tail-log backup is then used after the database restore to recover the transactions that were not included 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.

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 required if you are restoring a file from a non-primary 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, and any transactions in the tail of the log may be lost.

Point-in-time restore

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. 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 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 the Tail-log backup option is selected.

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 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 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. This option also creates a file with recovery changes. You can use the file to revert the recovery changes, if necessary.

[“Recovery operation options” on page 131](#) provides details on selecting recovery operations in different restore scenarios.

Disaster recovery

To ensure that you are sufficiently prepared for disaster recovery of a SQL Server environment, you must perform regular 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.

New features in this release

The following topics list the new features in this release.

New features in release 6.1 SP1

The Avamar Plug-in for SQL Server release 6.1 SP1 adds support for the following versions of SQL Server on Windows Server 2012:

- ◆ SQL Server 2012
- ◆ SQL Server 2008 Service Pack 3 or later
- ◆ SQL Server 2008 R2 Service Pack 1 or later

On Windows Server 2012, the SQL Server plug-in is supported only on a stand-alone server. Windows Server 2012 cluster support will be provided in a future release.

[“Supported configurations” on page 36](#) provides a complete list of supported SQL Server versions and operating system versions for this release of Avamar.

New features in release 6.1

The following new features were added to the Avamar Plug-in for SQL Server for release 6.1:

- ◆ **SQL Server 2012 support** — The SQL Server plug-in supports SQL Server 2012 on Windows Server 2008 R2 SP1 (64-bit) or Windows Server 2008 SP2 (32-bit or 64-bit), including support for AlwaysOn availability groups. [“SQL Server 2012 support and requirements” on page 32](#) provides details.
- ◆ **Microsoft .NET Framework 4 installation requirement** — Before you install the Avamar Plug-in for SQL Server, you must install Microsoft .NET Framework 4. Search the Microsoft Download Center for “Microsoft .NET Framework 4” to find downloads and additional information.
- ◆ **Improved backup and restore performance with multi-streaming** — Back up and restore data on the Avamar server or a Data Domain system by 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. [“Multi-streaming” on page 75](#) provides details.
- ◆ **Support for restoring filegroups and files** — You can now restore one or more filegroups or files to either the original location or to a different instance on either the same server or a different server. [“Restore” on page 25](#) provides details.
- ◆ **Support for restoring to an operating system file** — You can use the SQL Server plug-in to restore an instance or database to operating system files. This is useful if the SQL Server plug-in is not installed on the target server, or you want to use the standard SQL Server restore tools for features that are not provided by the Avamar Plug-in for SQL Server. [“Restoring to a file” on page 107](#) provides details.
- ◆ **Automatically skip or promote transaction log backups for databases with the simple recovery model** — If you are performing an incremental (transaction log) backup, then you can use the “For simple recovery model databases” plug-in option to specify how Avamar handles databases that use the simple recovery model, which does not support transaction log backups. Avamar can exclude databases with the simple

recovery model from the backup and write either an error or a warning message to the log. Alternatively, Avamar can perform a full backup instead of a transaction log backup. [“Backup options” on page 140](#) provides details.

- ◆ **Recovery operation selection** — You can control the recovery operation that occurs after the restore using restore plug-in options. The following recovery operations are supported:
 - RECOVERY
 - NORECOVERY
 - STANDBY

[“Recovery operation options” on page 131](#) provides details.
- ◆ **Automatic management of SQL services during system database restore** — Avamar can automatically stop and restart the necessary SQL services when you restore system databases when you enable the Manage SQL services automatically during restore checkbox in the restore plug-in options. [“System database restore options” on page 132](#) provides details.
- ◆ **Simplified database naming during restore** — When you restore SQL Server data from Avamar, the resulting file is named DESTINATION_DIR/INSTANCE/DATABASE/FILENAME, where DESTINATION_DIR is the target directory for the restore, INSTANCE is the name of the original instance for the restored data, DATABASE is the name of the database, and FILENAME is the name of the database files. If the data was backed up using multiple streams, then there will be one restored file for each stream.
- ◆ **User interface improvements** — The user interface for the backup and restore plug-in options has been changed to include grouping of related options and more descriptive option names. [Chapter 3, “Backup,”](#) and [Chapter 4, “Restore,”](#) provide details, including illustrations of the revised dialog boxes.
- ◆ **Command-line interface (CLI) support** — You can back up and restore SQL Server data from the command line using the SQL Server plug-in CLI. [Appendix B, “Command-Line Interface,”](#) provides details.

SQL Server 2012 support and requirements

The Avamar Plug-in for SQL Server adds support for SQL Server 2012 on the following operating systems:

- ◆ Windows Server 2008 R2 SP1 (64-bit)
- ◆ Windows Server 2008 SP2 (32-bit or 64-bit)
- ◆ Windows Server 2012

On Windows Server 2012, the SQL Server plug-in is supported only on a stand-alone server. Windows Server 2012 cluster support will be provided in a future release.

The SQL Server plug-in supports SQL Server 2012 in a cluster on Windows Server 2008 or Windows Server 2008 R2, including AlwaysOn availability groups. [“AlwaysOn availability groups” on page 17](#) provides details on how Avamar works with availability groups.

Keep in mind the following points when you use the Avamar Plug-in for SQL Server to back up and restore databases in SQL Server 2012:

- ◆ Before you perform a backup, you must add the Windows system service account to the SQL Server administrators group on each node in the cluster by performing the following steps:
 1. In Microsoft 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**.
- ◆ If you use AlwaysOn availability groups, Avamar can perform both on-demand and scheduled backups of databases in only the primary replica for an availability group. As a result, you should perform a full backup of the databases in an availability group after it fails over from one node to another.
- ◆ When you back up or restore databases in a SQL Server 2012 AlwaysOn availability group, select the hostname of the physical node that is hosting the primary replica as the client. Backup and restore fails if you select the virtual server name in the cluster or the availability group name as the client.
- ◆ If you use AlwaysOn availability groups, you can use the backup from the primary replica to restore the databases in either the primary replica or the secondary replicas. [“Restoring a database in an AlwaysOn availability group” on page 127](#) provides details.

Known limitations

This topic discusses known limitations of the Avamar Plug-in for SQL Server.

Avamar plug-in and server version compatibility issues

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

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

Backing up databases with FILESTREAM data is not supported

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 cannot back up SQL Server databases with FILESTREAM data.

You can use SQL Server management tools to back up databases with FILE STREAM data. The Microsoft website provides additional details on the FILESTREAM feature and how to back up and restore databases that store BLOB data on the file system.

Different database sizes appear in Avamar Administrator

Due to inherent limitations in the way that database sizes are calculated, database sizes shown in the Avamar Administrator Backup and Restore window Backup pane are generally accurate, while database sizes shown in the Restore pane can be significantly smaller. This is because the exact size of a restored database cannot be accurately known until after the restore operation completes.

Viewing partial results after a browse timeout is not supported

If there are a significant number of databases on an SQL server and you are browsing the Backup and Restore window for a database to back up or restore, then the Avamar Management Console Server (MCS) may not be able to display all of the 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 choose to view partial results, no entries are listed. The Avamar Plug-in for SQL Server does not support this option.

Backing up the Resource database is not supported

The SQL Server Resource database is a read-only database that contains all the system objects that are included with SQL Server. Avamar 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.

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 time that 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 necessary.

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

CHAPTER 2

Installation

The following topics describe how to install and configure Avamar Plug-in for SQL Server software:

- ◆ System requirements 36
- ◆ Installation road map 40
- ◆ Installing the Avamar Client for Windows 41
- ◆ Installing the Avamar Plug-in for SQL Server 43
- ◆ Registering the client 44
- ◆ Installing the Avamar Cluster Client 45
- ◆ Verifying the configuration 47
- ◆ Upgrade 48
- ◆ Uninstall 50

System requirements

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

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/products>.

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

- ◆ “Supported configurations” on page 36
- ◆ “Hardware requirements” on page 37
- ◆ “SQL Server requirements” on page 37
- ◆ “Security requirements” on page 39
- ◆ “Avamar server dependencies” on page 39

Supported configurations

The following table lists the SQL Server versions and host operating systems that the Avamar Plug-in for SQL Server supports.

Table 3 Supported host operating systems for SQL servers

SQL Server version	Supported host operating systems
2012	<ul style="list-style-type: none"> • Windows Server 2012 • Windows Server 2008 R2 SP1 (64-bit) • Windows Server 2008 SP2 (32-bit or 64-bit)
2008 R2 SP1 or later	<ul style="list-style-type: none"> • Windows Server 2008 R2 or later (64-bit) • Windows Server 2008 SP2 (32-bit or 64-bit) • Windows Server 2003 SP2 (32-bit or 64-bit)
2008 SP3 or later	Windows Server 2012
2008 SP1 or later	<ul style="list-style-type: none"> • Windows Server 2008 R2 or later (64-bit) • Windows Server 2008 SP1 or later (32-bit or 64-bit) • Windows Server 2003 SP1 or later (32-bit or 64-bit)
2005 SP3 or later	<ul style="list-style-type: none"> • Windows Server 2008 R2 or later (64-bit) • Windows Server 2008 SP1 or later (32-bit or 64-bit) • Windows Server 2003 SP1 or later (32-bit or 64-bit)

Various clustered configurations are also supported. The *EMC Avamar Compatibility and Interoperability Matrix* provides the most up-to-date information about supported configurations.

Note: The SQL Server plug-in is supported on Windows Server 2012 only on a stand-alone server. Windows Server 2012 cluster support will be provided in a future release.

Before you install the Avamar Plug-in for SQL Server, you must install Microsoft .NET Framework 4. Otherwise, the installation cannot proceed. Search the Microsoft Download Center for “Microsoft .NET Framework 4” to find downloads and additional information.

Hardware requirements

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

Table 4 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 used for local cache files.
Network interface	10BaseT minimum; 100BaseT or higher recommended, configured with the latest drivers for the platform.

SQL Server requirements

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

SQL Server feature 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.

Installation of Management Tools and the SQL Client Connectivity SDK is also required when you install SQL Server.

Database name requirements

Characters used in database names are restricted to valid file name characters. You should particularly avoid any of the following characters, which are known to interfere with proper operation of the Avamar Plug-in for SQL Server: asterisk (*), forward slash (/), backward slash (\), colon (:), semicolon (;), question mark (?), right angle bracket (>), left angle bracket (<), or vertical bar (|).

In addition, 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 attempt to open the folder to which you restore the 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.

Alias requirements

A SQL Server alias is a name, configured on the client computer that points to a SQL 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, if it is improperly used, it can result in connectivity issues that are difficult to isolate and troubleshoot.

If you plan to use a SQL Server alias, then you should review the requirements and configuration steps on the Microsoft website for setting up an alias. In addition, 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

Follow the requirements in the following topics when you back up and restore databases in SQL Server 2012 AlwaysOn availability groups.

Backup requirements for AlwaysOn availability groups

You can only back up databases in the primary replica for an availability group. As a result, you should perform a full backup of the databases in an availability group after it fails over from one node to another.

When you back up a database in an availability group, select the hostname of the physical node that is hosting the primary replica as the client to back up. Determine the node that is hosting the primary replica by using one of the following methods:

- ◆ Open Failover Cluster Manager and view which node is the current owner of the availability group.
- ◆ Open SQL Server Management Studio and view the settings for the availability group.

If you do not know which node is hosting the primary replica at the time of the backup, you can perform a backup of all replicas. Specify the hostname for all physical nodes that might host the availability group. The backup succeeds on the node that is hosting the primary replica at the time of the backup and fails on the nodes that are hosting the secondary replicas.

Restore requirements for AlwaysOn availability groups

When you restore a database in the primary replica, remove the database from the availability group, select the hostname of the physical node that is hosting the primary replica as the Avamar client from which to restore, and then use the Avamar Plug-in for SQL Server to restore the database to its original location, as described in [“Restoring to the original location” on page 94](#). Then add the database back to the availability group.

NOTICE

After you remove the database from the availability group on the primary replica, the corresponding database on the secondary replicas are in a restoring state. There are two ways to restore the databases on the secondary replicas as part of the availability group. You can either delete the databases on the secondary replicas and then automatically re-create and synchronize the databases when you add the database back to the availability group, or you can 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. The SQL Server 2012 documentation available on the Microsoft website provides details.

When you restore a database in a secondary replica, remove the database from the availability group, select the hostname of the physical node that is hosting the primary replica as the Avamar client from which to restore, and then use the Avamar Plug-in for SQL Server to restore the database to the corresponding SQL Server instance on the other

cluster node, as described in [“Restoring to a different instance” on page 101](#). During the restore, select NORECOVERY for the recovery operation in the plug-in options. After the restore, join the database to the to the availability group as described in the SQL Server 2012 documentation on the Microsoft website.

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.
- ◆ You must perform backups of only the principal database, not the mirrors.
- ◆ You must 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.
- ◆ Extra care is needed by the backup and database administrators, as 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.

Security requirements

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

Avamar server dependencies

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

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 later on both the client and server.

If you must upgrade the Avamar server to release 6.1 or later but you cannot upgrade the SQL Server plug-in to release 6.1 or later, then use the release 6.0 plug-in catalog on the Avamar server.

Installation road map

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

Standalone installation road map

To install the necessary Avamar software to back up and restore a SQL server environment on a standalone server:

1. Install and configure Microsoft SQL Server.
2. Install Microsoft .NET Framework 4, which 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:
 - a. In Microsoft SQL Server Management Studio, expand the **Security** node and then the **Logins** node for the instance in the left pane.
 - b. Right-click the **NT AUTHORITY\SYSTEM** account and select **Properties**.
The Login Properties dialog box appears.
 - c. Select the **Server Roles** page from the list in the left pane.
 - d. In the right pane, select the checkbox next to the **sysadmin** user.
 - e. Click **OK**.
4. Install the Avamar Client for Windows on the SQL server as described in ["Installing the Avamar Client for Windows" on page 41](#).
5. Install the Avamar Plug-in for SQL Server on the SQL server as described in ["Installing the Avamar Plug-in for SQL Server" on page 43](#).
6. Register the Avamar Plug-in for SQL Server with the Avamar server as described in ["Registering the client" on page 44](#).
7. (Optional) Use the Avamar Config Checker for Microsoft Windows to verify if the environment is properly configured to perform backups and restores as described in ["Verifying the configuration" on page 47](#).

Cluster installation road map

To install the necessary Avamar software to back up and restore a SQL server environment in a cluster environment:

1. Install and prepare the cluster.
2. Install and configure Microsoft SQL Server in the cluster.
3. Install Microsoft .NET Framework 4 on each node in the cluster. 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.

4. If you are using Microsoft SQL Server 2012, then add the Windows system service account to the SQL Server administrators group on each node in the cluster:
 - a. In Microsoft SQL Server Management Studio, expand the **Security** node and then the **Logins** node for the instance in the left pane.
 - b. Right-click the **NT AUTHORITY\SYSTEM** account and select **Properties**.
The Login Properties dialog box appears.
 - c. Select the **Server Roles** page from the list in the left pane.
 - d. In the right pane, select the checkbox next to the **sysadmin** user.
 - e. Click **OK**.
5. Install the Avamar Client for Windows on all nodes in the cluster as described in [“Installing the Avamar Client for Windows” on page 41](#).
Install each client to the same directory on each cluster node. For example, if you install the Avamar Client for Windows to C:\AVS1 on the first node, then you must install the Avamar Client for Windows to the same directory, C:\AVS1, on all other nodes in the cluster.
6. Install the Avamar Plug-in for SQL Server to the same directory on each cluster node as described in [“Installing the Avamar Plug-in for SQL Server” on page 43](#).
7. Register each node in the cluster with the Avamar server as described in [“Registering the client” on page 44](#).
8. If SQL Server is configured as a Failover Cluster Instance (FCI), then use the Cluster Configuration Tool to install the Avamar Cluster Client on the active node in an active/passive configuration or on both active nodes in an active/active configuration, as described in [“Installing the Avamar Cluster Client” on page 45](#).

Note: You do not need to use the Cluster Configuration Tool to install the Avamar Cluster Client if you are only backing up databases in an AlwaysOn availability group.

9. (Optional) Use the Avamar Config Checker for Microsoft Windows to verify if the environment is properly configured to perform backups and restores as described in [“Verifying the configuration” on page 47](#).

Installing the Avamar Client for Windows

To install the Avamar Client for Windows:

1. Log in to the client computer that hosts SQL Server with an account that has Administrator privileges.
2. Open a web browser and go to the following URL:

http://AVAMARSERVER

where AVAMARSERVER is the network hostname (as defined in DNS) or IP address of the Avamar system.

You are automatically redirected to the Avamar secure web server.

Depending on the browser security settings, a security alert dialog box might appear.

3. If a security alert dialog box appears, click **Yes** or **OK** to allow redirection to the Avamar secure web server.

The Secure Log On page appears.

4. Page down and then click the **Documents and Downloads** hyperlink.

The Documents and Downloads page appears.

5. Click the correct operating system hyperlink for the client computer.

A directory listing appears.

6. Select the install package for the correct Windows version:

- **AvamarClient-windows-x86-VERSION.msi** (32-bit)
- **AvamarClient-windows-x86_64-VERSION.msi** (64-bit)

where VERSION is the client version that you are installing.

The browser prompts you to either run the file or save it to the local computer. You can use either method unless the local computer has the User Account Control (UAC) feature, in which case you must download the install package to a temporary directory.

UAC is designed to provide additional operating system security by preventing software from being installed with administrator privileges, unless an administrator authorizes the elevated privileges.

7. Start the installer, using one of the following set of steps, depending on whether UAC is enabled:

- If UAC is disabled, open the file in place on the server, or download the file to a temporary directory on the computer, and then open the file.
- 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 directory that contains the installation package.

- c. On a single line, type one of the following commands, depending on the Windows version, to launch the installer:

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

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

where VERSION is the version number of the installation package.

The installation wizard appears.

8. Follow the instructions to proceed through the wizard.

9. Click **Finish**.

Installing the Avamar Plug-in for SQL Server

To install the Avamar Plug-in for SQL Server:

1. Log in to the client computer that hosts SQL Server with an account that has Administrator privileges.

2. Open a web browser and go to the following URL:

http://AVAMARSERVER

where AVAMARSERVER is the network hostname (as defined in DNS) or IP address of the Avamar system.

You are automatically redirected to the Avamar secure web server.

Depending on the browser security settings, a security alert dialog box might appear.

3. If a security alert dialog box appears, click **Yes** or **OK** to allow redirection to the Avamar secure web server.

The Secure Log On page appears.

4. Page down and then click the **Documents and Downloads** hyperlink.

The Documents and Downloads page appears.

5. Click the correct operating system hyperlink for the client computer.

A directory listing appears.

6. Double-click the Avamar Plug-in for SQL Server install package for the correct Windows version:

- **AvamarSQL-windows-x86-VERSION.msi** (32-bit)
- **AvamarSQL-windows-x86_64-VERSION.msi** (64-bit).

where VERSION is the client version that you are installing.

The browser prompts you to either run the file or save it to the local computer.

7. Start the installer, using one of the following set of steps, depending on whether UAC is enabled:

- If UAC is disabled, open the file in place on the server, or download the file to a temporary directory on the computer, and then open the file.
- 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 directory that contains the installation package.

- c. On a single line, type one of the following commands, depending on the Windows version, to launch the installer:

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

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

where VERSION is the version number of the available install package.

The installation wizard appears.

8. Follow the instructions to proceed through the wizard.
9. Click **Finish**.

Registering the client

Before you can back up or restore SQL Server data, you must register the SQL Server client with the Avamar server.

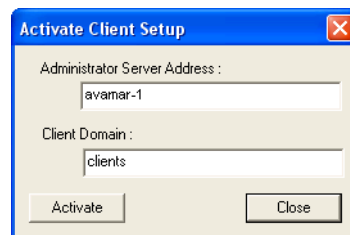
In a cluster environment, register all SQL Server nodes in the cluster as Avamar clients.

To register the client:



1. Log in to the SQL server.
2. Right-click the Avamar client system tray icon.
A menu appears.
3. Select **Activate**.

The Activate Client Setup dialog box appears.



4. In the **Administrator Server Address** box, type the network hostname of the Avamar server as defined in DNS.
5. In the **Client Domain** box, type the name of the Avamar domain for the client.

The default domain is “clients.” Consult the Avamar system administrator for the appropriate domain or subdomain to use.

To type a subdomain, do not use a slash (/) as the first character. If you use a slash, an error occurs and you cannot register the client.

6. Click **Activate**.
A confirmation message appears.
7. Click **OK**.

Installing the Avamar Cluster Client

If SQL Server is configured as a Failover Cluster Instance (FCI), then use the Avamar Cluster Configuration Tool to install the Avamar Cluster Client on the active node of an active/passive cluster or on both active nodes in an active/active cluster, and to register the virtual server as a client on the Avamar server.

Note: You do not need to use the Cluster Configuration Tool to install the Avamar Cluster Client if you are only backing up databases in an AlwaysOn availability group.

To install the Avamar Cluster Client:

1. Ensure that you have performed the following installation steps:
 - a. Installed the Avamar Client for Windows to the same directory on all nodes in the cluster.
 - b. Installed the Avamar Plug-in for SQL Server to the same directory on all nodes in the cluster.
 - c. Activated all physical nodes as clients on the Avamar server.
2. Log in to the active node in the cluster with an account that has domain administrator privileges and is a member of the local Administrators group on each cluster node.
3. From the avs\bin directory, run **AvamarClusterConfiguration.exe**.

The Setup - Cluster Backup Agent for Windows dialog box appears.

4. From the **Choose a Group for Install or Update** list, select the cluster group that contains the SQL server on which to install the Avamar Cluster Client.

5. From the **Network name for backup agent communication** list, select the network name used by SQL Server for Avamar client-server communication.
The name is listed in the Network Name dependency of the SQL Server resource.
6. From the **Shared volume for backup agent configuration and log files** list, select the shared external storage drive on which to install the Avamar Cluster Client.
7. In the **Var folder** box, type or browse to the UNC path for a shared network folder or shared volume on which to store the Avamar Cluster Client configuration and log files. To create a shared folder, click **Share Dir**.
If a shared volume is available, then a path appears automatically in the box.
All nodes in the cluster must be able to access and write to the location.
Selecting the volume that is owned by the cluster is recommended instead of a remote path on a network.
8. Choose whether to register the client associated with the cluster group with the Avamar server by selecting or clearing the **Do not register client now** checkbox.
If you select the checkbox, then you must manually register and activate the client before you can perform backups.
9. In the **Administrator server hostname or IP address** box, specify the network hostname as defined in DNS or the IP address of the Avamar server.
10. In the **Administrator server listen port** box, specify the data port for Avamar client-server communication.
Unless you use a different data port, leave the default value of 28001.
11. In the **Backup domain for this client** box, specify the Avamar domain for the client associated with the cluster group.
The default domain is the “clients” domain. Consult the Avamar system administrator for the domain to use for the client. If you type a subdomain, such as clients/MyClients, do not include a slash (/) as the first character. Otherwise, an error occurs and prevents the registration of the client.
12. To specify a name for the Avamar Cluster Client instead of using the default value of the cluster group name, type the name in the **Override automatic backup client name with this name** box.
Using the cluster group name is recommended.
13. In the **Plug-ins** list, ensure that the list contains the plug-ins that can run in a cluster.
You should not need to change the default values in the list. However, if you do not want Avamar to use a particular plug-in, select the plug-in and click **Remove**. If you remove a plug-in and want to return it to the list, click **Set All**.
14. In the **Cluster Nodes** list, ensure that the list contains the cluster nodes on which to install the Avamar Cluster Client.
The list automatically contains all physical nodes in the cluster. You should not need to change the default list. However, to prevent the installation of the Avamar Cluster Client on a node, select the node and click **Remove**. If you remove a node and want to return it to the list, click **Set All**.

15. Click **Configure**.

If you specified a UNC path in the Var folder box, you are prompted for credentials to run the Avamar service. The credentials should have the ability to run as a service on the specified server as either a local or domain account.

When installation is complete, a confirmation message box appears and lists the plug-ins that were configured.

16. Click **OK**.

The setup dialog box appears with the Install button dimmed.

17. If the Var folder is on a network share, start the backup cluster agent from an account that has full access permissions to the folder where Avamar log files are written.

Verifying the configuration

You can use the Avamar Config Checker for Microsoft Windows to verify if the environment is properly configured for Avamar backup and recovery. You can use the Config Checker either before or after you install an Avamar client or plug-in.

The Config Checker checks and captures configuration errors that may lead to failure in Avamar installation, data backup, or data recovery for:

- ◆ The application host (operating system environment)
- ◆ Microsoft SharePoint
- ◆ Microsoft Exchange
- ◆ Microsoft SQL Server
- ◆ Microsoft Hyper-V

The Config Checker is supported only on English language operating systems.

To install the Avamar Config Checker:

1. Log in to the client computer.
2. Open a web browser and type the following URL:

http://AVAMARSERVER

where AVAMARSERVER is the network hostname as defined in DNS or the IP address of the Avamar server.

You are redirected to the Avamar secure web server.

3. If a security alert dialog box appears due to browser security settings, click **Yes** or **OK** to allow redirection to the Avamar secure web server.

The Secure Log On page appears.

4. Scroll down and then click the **Documents and Downloads** hyperlink.

The Documents and Downloads page appears.

5. Locate the link for the correct operating system for the client computer, and then click the corresponding install package:
 - **Windows for x86 (32 bit) > Avamar_ConfigChecker_win_x86.zip**
 - **Windows for x86 (64 bit) > Avamar_ConfigChecker_win_x64.zip**
6. Save the install package to a temporary directory, unzip the folder, and then run the installation program.

The *EMC Avamar Config Checker for Microsoft Windows Supplemental Notes*, available on the EMC Online Support at <https://support.EMC.com/products>, provides complete instructions to install, run, and uninstall the Config Checker, as well as examples of HTML results files.

Upgrade

The following topics explain how to upgrade Avamar components in a SQL Server environment.

Upgrading the Avamar Plug-in for SQL Server

To upgrade the Avamar Plug-in for SQL Server to version 6.1 from an earlier version, run the installation wizard for version 6.1 on the server. You do not need to uninstall earlier versions of the plug-in before you install a new version.

NOTICE

If you upgrade the Avamar Plug-in for SQL Server to release 6.1, then you must also upgrade the Avamar server software to release 6.1. Do not use a SQL Server plug-in running release 6.0 or earlier to back up to or restore from an Avamar server running release 6.1. Also, do not use a SQL Server plug-in running release 6.1 to back up to or restore from an Avamar server running release 6.0 or earlier. The user interface and underlying code for several features, including point-in-time restore and redirected restore, has changed in release 6.1. The changes require release 6.1 on both the client and server.

In addition, you should perform a full backup after the upgrade to ensure that you have a current full backup with the newer version. You may also want to perform a differential or transaction log (incremental) backup to preserve the recovery model in use.

Upgrading to Avamar Cluster Client

To uninstall an older version of Windows Cluster Client and upgrade to Avamar Cluster Client:

1. Set the cluster offline using the appropriate cluster administration tool:
 - On Windows Server 2008, use **Failover Cluster Management**.
 - On Windows Server 2003, use **Cluster Administrator**.
2. Right-click the **Backup Agent** service and set it offline.

3. On the passive node, use the Windows Server 2003 Add/Remove Programs feature or the Windows Server 2008 Programs and Features feature to uninstall the following Avamar clients and plug-ins:
 - Windows client
 - SQL Server plug-in
4. On the passive node, follow the steps in [“Uninstalling the Avamar Cluster Client” on page 50](#) to uninstall the Windows Cluster Client.
5. On the active node, use the Windows Server 2003 Add/Remove Programs feature or the Windows Server 2008 Programs and Features feature to uninstall the following Avamar clients and plug-ins:
 - Windows client
 - SQL Server plug-in
6. On the active node, follow the steps in [“Uninstalling the Avamar Cluster Client” on page 50](#) to uninstall the Windows Cluster Client.
7. Use the cluster administration tool to delete the Backup Agent resource.
8. Install the new version of the Avamar Client for Windows in the same Var folder on all nodes as described in [“Installing the Avamar Client for Windows” on page 41](#).

When you install the Windows client (and later the SQL Server plug-in) in the same Var folder as the previous Avamar installation, the config and log files are still available there because the uninstall process does not remove this folder.
9. Install the new version of the Avamar Plug-in for SQL Server in the same Var folder on all nodes as described in [“Installing the Avamar Plug-in for SQL Server” on page 43](#).
10. Install the Avamar Cluster Client on the active node using the instructions in [“Installing the Avamar Cluster Client” on page 45](#).

Uninstall

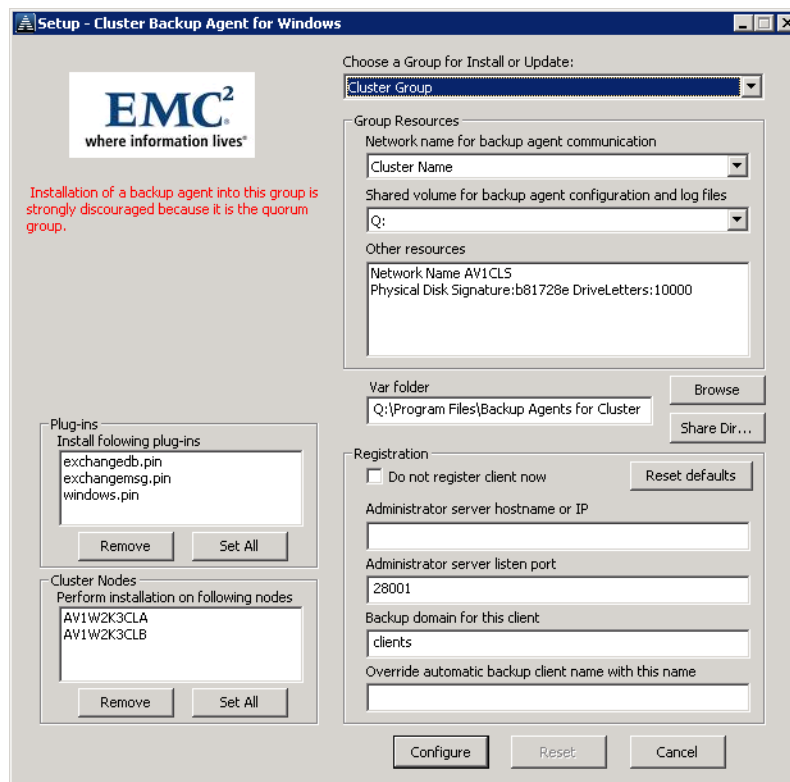
The following topics explain how to uninstall Avamar components in a SQL Server environment.

Uninstalling the Avamar Cluster Client

To uninstall the Avamar Cluster Client:

1. Ensure that all cluster nodes are operational.
2. Log in to the active node in the Windows cluster with an account that has Administrator privileges.
3. In the bin directory of the Avamar installation directory, which is usually C:\Program Files\avs\bin, run **AvamarClusterConfiguration.exe**.

The Setup - Cluster Backup Agent for Windows dialog box appears.



4. From the **Choose a Group for Install or Update** list, select the cluster group from which to uninstall the Avamar Cluster Client.
5. Click **Reset**.

Uninstalling the Avamar Plug-in for SQL Server

To uninstall the SQL Server plug-in, use the Add/Remove Programs feature in Windows 2003 or the Programs and Features feature in Windows Server 2008 or Windows Server 2012. These features are available in the Control Panel.

Uninstalling the Avamar Client for Windows

To uninstall the Windows client, use the Add/Remove Programs feature in Windows 2003 or the Programs and Features feature in Windows Server 2008 or Windows Server 2012. These features are available in the Control Panel.

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:

- ◆ [On-demand backup](#) 54
- ◆ [Scheduled backups](#)..... 61
- ◆ [Multi-streaming](#) 75

On-demand backup



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 and Restore window appears.

2. In the clients tree, select the SQL server. Keep the following points in mind when you select a client:
 - If you are backing up databases in a SQL Server 2012 AlwaysOn availability group, then select the hostname of the physical node that is hosting the primary replica. [“Backup requirements for AlwaysOn availability groups” on page 38](#) explains how to determine which node is hosting the primary replica.
 - If SQL Server is configured as a Failover Cluster Instance (FCI) and you are backing up databases on shared storage in the cluster, then select the Avamar Cluster Client.
 - You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
3. Click the **Select for Backup** tab.

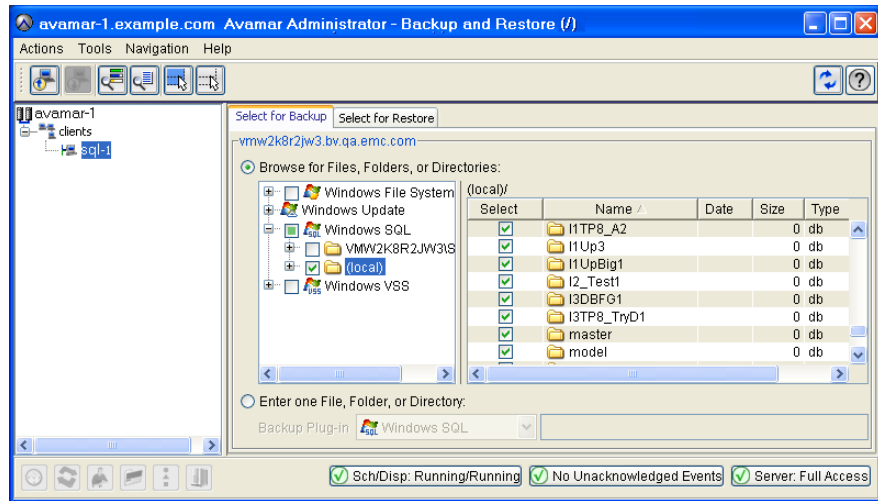
A list of plug-ins installed on the selected client appears in the left pane of the Select for Backup tab.

4. Select the data to back up:
 - To back up all SQL Server data on the client, select the checkbox next to the **Windows SQL** plug-in node in the left pane of the **Select for Backup** tab.
 - To browse to one or more instances or databases to back up:
 - a. Select **Browse for Files, Folders, or Directories**.
 - b. Expand the **Windows SQL** plug-in node in the left pane of the **Select for Backup** tab.
 - c. Under the **Windows SQL** plug-in node, select the database instance that contains the databases to back up.

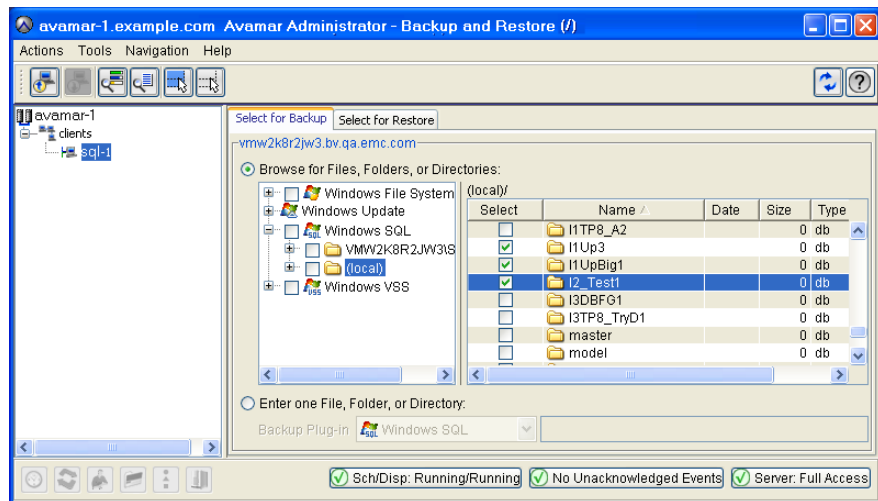
A list of the databases for that instance appears in the right pane of the Select for Backup tab.

d. Select the data to backup.

To back up all databases in the instance, select the checkbox next to the instance in the left pane of the **Select for Backup** tab.



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



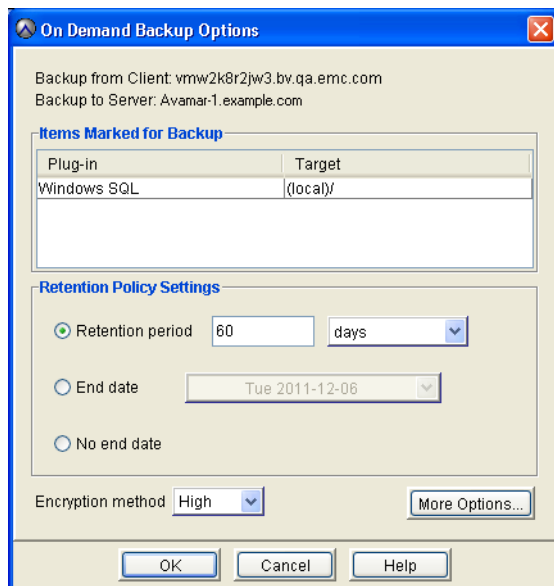
- To type the instance or database to back up:
 - a. Select **Enter one File, Folder, or Directory**.
 - b. From the **Backup Plug-in** list, select **Windows SQL**.
 - c. In the box next to the **Backup Plug-in** list, type the instance or database:
 - To back up the local instance, type (local).
 - 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 named local, type CLIENT/DATABASE/.

where CLIENT is the name of the SQL server, INSTANCE is the name of the named instance, and DATABASE is the name of the database.

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

The On Demand Backup Options dialog box appears.



6. Select the backup retention setting in **Retention Policy Settings**:

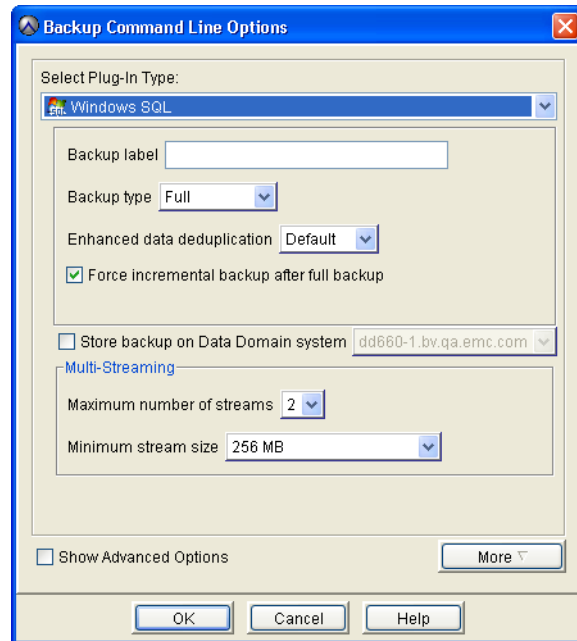
- To automatically delete this backup from the Avamar server after a specific amount of time, select **Retention period** and then specify the number of days, weeks, months, or years for the retention period.
- To automatically delete this backup from the Avamar server on a specific calendar date, select **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**.

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

The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

8. Click **More Options**.

The Backup Command Line Options dialog box appears, as shown in the following figure.



9. Set the plug-in options:

- a. To include a descriptive label for the backup, type the label in the **Backup label** box.
- b. From the **Backup type** list, choose the type of backup:
 - Select **Full** to back up the entire database, including all objects, system tables, and data.
 - Select **Differential** to back up any data that has changed since the last full backup.
 - Select **Incremental** to back up only the transaction logs.
- c. From the **Enhanced data deduplication** list, choose whether to use enhanced data deduplication, which typically reduces the amount of client data that must be sent to the server but requires additional client CPU resources:
 - To use the global enhanced data deduplication setting already set on the server, select **Default**.
 - To disable enhanced data deduplication, select **Disabled**.
 - To enable enhanced data deduplication, select **Enabled**.
- d. Choose whether to force a transaction log (incremental) backup that contains transactions that occur between full backups by selecting or clearing the **Force incremental backup after full backup** checkbox.

Forcing a transaction log backup between full backups ensures that a point-in-time recovery to a point in time between the two full backups can occur, if necessary.

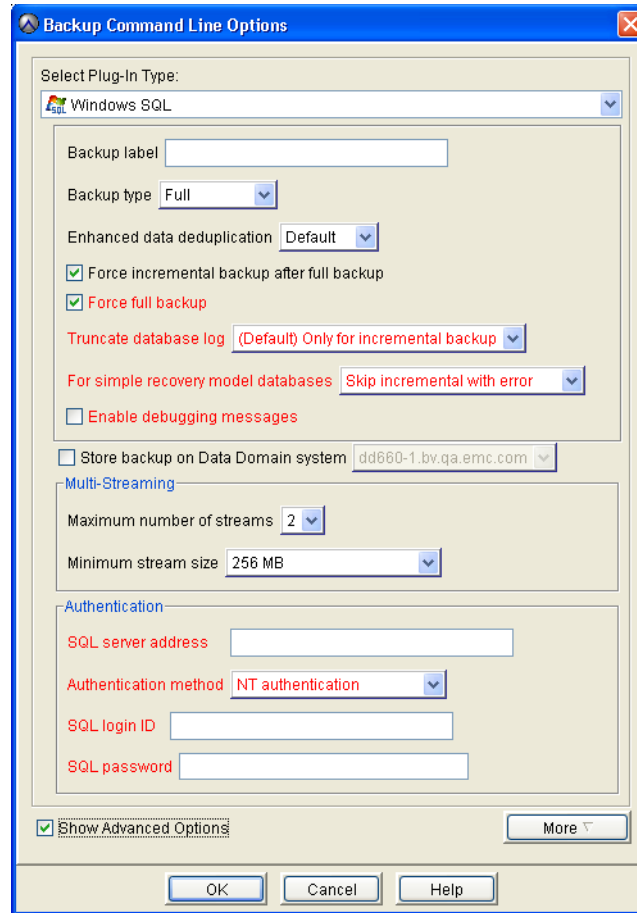
NOTICE

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 in the advanced options to control how Avamar handles transaction log backups of databases that use the simple recovery model.

- e. To store the backup on a Data Domain system instead of the Avamar server, select **Store backup on Data Domain system** and select the Data Domain system from the list.
- f. To specify the maximum number of streams to use for backup and restore, select a value from the **Maximum number of streams** list. [“Multi-streaming” on page 75](#) provides details on multi-streaming.
- g. To specify the minimum size of each data stream, select a value from the **Minimum stream size** list.

- h. To specify advanced backup options, select the **Show Advanced Options** checkbox. Otherwise, proceed to [step 10](#).

Additional options appear in red in the Backup Command Line Options dialog box, as shown in the following figure.



- i. Select or clear the **Force full backup** checkbox to specify whether to perform a full backup when Avamar detects a log gap or when there is no previous full backup, from which a transaction log (incremental) or differential backup can be applied, on the server for the database. Effectively, this option automates taking a full backup when necessary.

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 in the event that no existing full backup is present on the Avamar server.

- j. Select an option from the **Truncate database log** list to control database transaction log truncation behavior:
 - Select **(Default) Only for incremental backup** to truncate the log if the backup type is set to incremental (transaction log). No log truncation occurs if the backup type is full or differential.
 - Select **For all backup types** to truncate the log regardless of the backup type.

NOTICE

This setting breaks the chain of log backups and should not be used unless the backup type is set to incremental (transaction log).

- Select **Never** if you do not want to truncate the log under any circumstances.
- k. If you are performing an incremental (transaction log) backup and you have selected databases that use the simple recovery model, which does not support transaction log backups, then specify how Avamar handles the databases by selecting one of the following options from the **For simple recovery model databases** list:
 - **Skip incremental with error** — If you select databases that use the simple recovery model and databases that use other recovery models, then Avamar excludes the databases with the simple recovery model from the backup and writes an error message to the log. The backup completes with exceptions. If you select only databases that use the simple recovery model, then the backup fails.
 - **Skip incremental with warning** — If you select databases that use the simple recovery model and databases that use other recovery models, then Avamar excludes databases with the simple recovery model from the backup and writes a warning to the log for each database that uses the simple recovery model. The backup completes successfully. If you select only databases that use the simple recovery model, then the backup fails.
 - **Promote incremental to full** — Avamar performs a full backup instead of a transaction log backup for databases that use the simple recovery model.
- l. Choose whether to write maximum information to log files by selecting or clearing the **Enable debugging messages** checkbox. If selected, very large log files are created.
- m. (Optional) Type the hostname or IP address of the SQL server in the **SQL server address** box.
- n. Choose whether to use NT authentication or SQL Server authentication to connect to SQL Server by selecting a value from the **Authentication method** list.
- o. If you select the SQL Server authentication method, type the SQL Server login ID and password in the **SQL login ID** and **SQL password** boxes, respectively.

[Appendix A, “Plug-in Options,”](#) provides additional information on the backup plug-in options.

10. Click **OK** on the **Backup Command Line Options** dialog box.

11. Click **OK** on the **On Demand Backup Options** dialog box.

The On Demand Backup Request dialog box indicates that the backup was initiated.

12. Click **Close**.

Scheduled backups

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

1. Create a dataset for the backups, as discussed in [“Creating a dataset” on page 61](#).
2. Create a group for the backups, as discussed in [“Creating a group” on page 71](#). 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, as discussed in [“Enabling scheduled backups” on page 75](#).

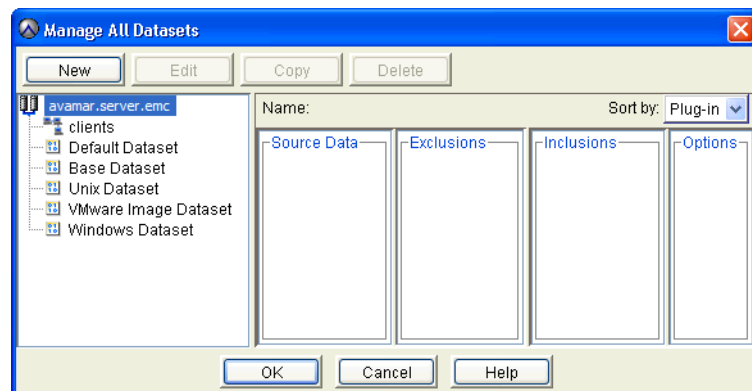
A thorough discussion of groups, group policy, datasets, schedules, and retention policies is beyond the scope of this guide. The *EMC Avamar Administration Guide* provides additional information.

Creating a dataset

To create a dataset for scheduled backups:

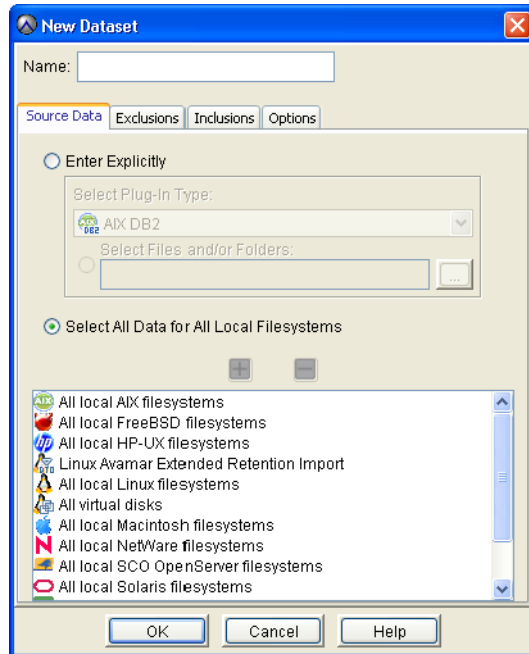
1. In Avamar Administrator, select **Tools > Manage Datasets**.

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.

Do not use any of the following characters in the dataset name:
~!@\$%^&{}|,;#\\:*?<>'""&.

4. On the **Source Data** tab, choose the data to include in the the dataset:

- To include data from all plug-ins installed on the client, select **Select All Data for All Local File Systems**.
- To include data from only specific plug-ins installed on the client:
 - a. Select **Enter Explicitly**.
 - b. To remove a plug-in from the dataset, select the plug-in from the list in the bottom portion of the **New Dataset** dialog box, and then click - (**Remove From List**). Repeat this step as necessary.
 - c. To add a plug-in to the dataset, select the plug-in from the **Select Plug-In Type** list, and then click + (**Add to List**). Repeat this step as necessary.

NOTICE

All Avamar plug-ins are listed in the **Select Plug-In Type** list, but only plug-ins installed on the client are available to add to the dataset.

- To limit the dataset to individual SQL Server instances or databases:

- Select **Enter Explicitly**.
- Click **... (Browse for files and/or folders)**.

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

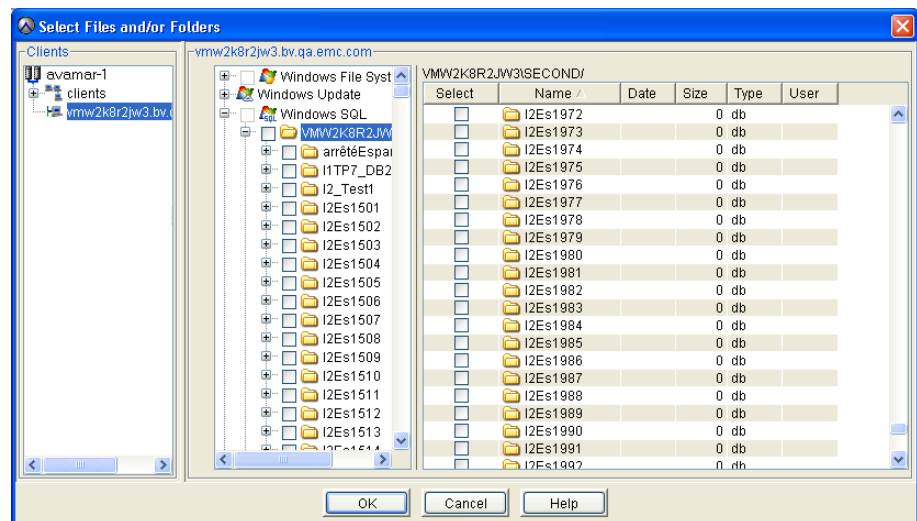
- Select the client from the **Clients** tree in the left pane.

If the dataset is for backups of databases in a SQL Server 2012 AlwaysOn availability group, then select the hostname of the physical node that is hosting the primary replica. “[Backup requirements for AlwaysOn availability groups](#)” on page 38 explains how to determine which node is hosting the primary replica.

If SQL Server is configured as a Failover Cluster Instance (FCI) and the dataset is for backups of databases on shared storage in the cluster, then select the Avamar Cluster Client.

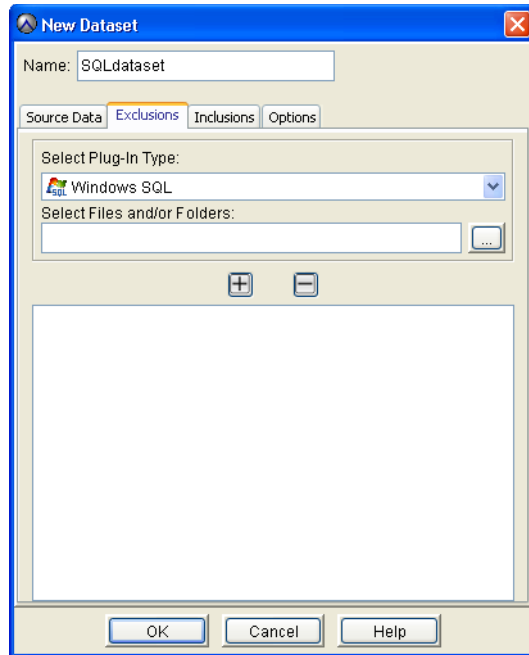
- In the middle pane of the **Select Files And/Or Folders** dialog box, expand the **Windows SQL** plug-in node.
- Under the **Windows SQL** plug-in node, select the database instance that contains the databases to include.

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



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

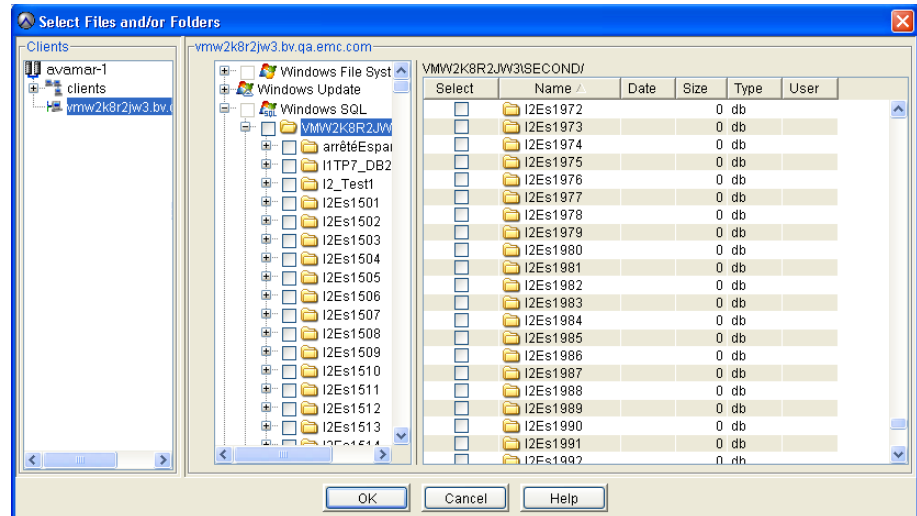
5. (Optional) Click the **Exclusions** tab and define databases or instances to exclude from the dataset:
 - a. From the **Select Plug-In Type** list, select **Windows SQL**, as shown in the following figure.



- b. Click ... to browse to the instances or databases to exclude.
The Select Files And/Or Folders dialog box appears.
- c. Select the client from the **Clients** tree in the left pane.
- d. In the middle pane of the **Select Files And/Or Folders** dialog box, expand the **Windows SQL** plug-in node.

- e. Under the **Windows SQL** plug-in node, select the database instance that contains the databases to exclude.

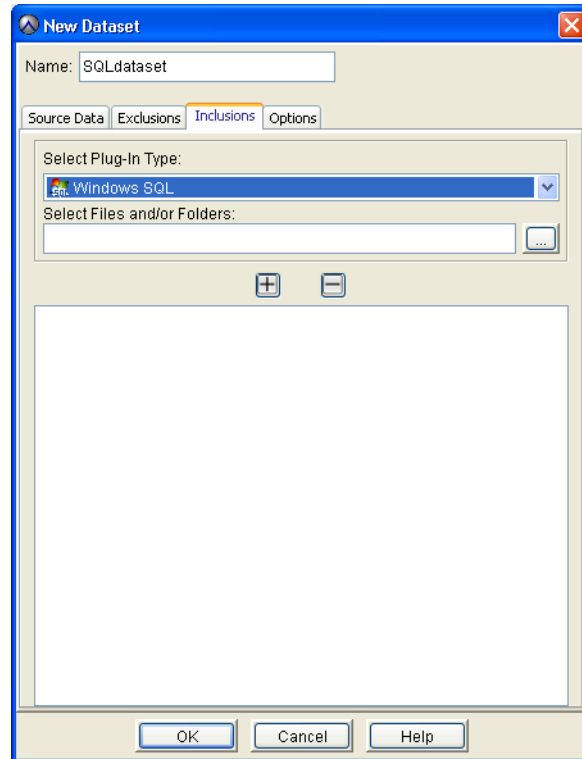
A list of the databases for that instance appears in the right pane of the Select Files And/Or Folders dialog box.



- f. 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.
- g. Click **OK**.

The selected instances or databases are listed on the Exclusions tab.

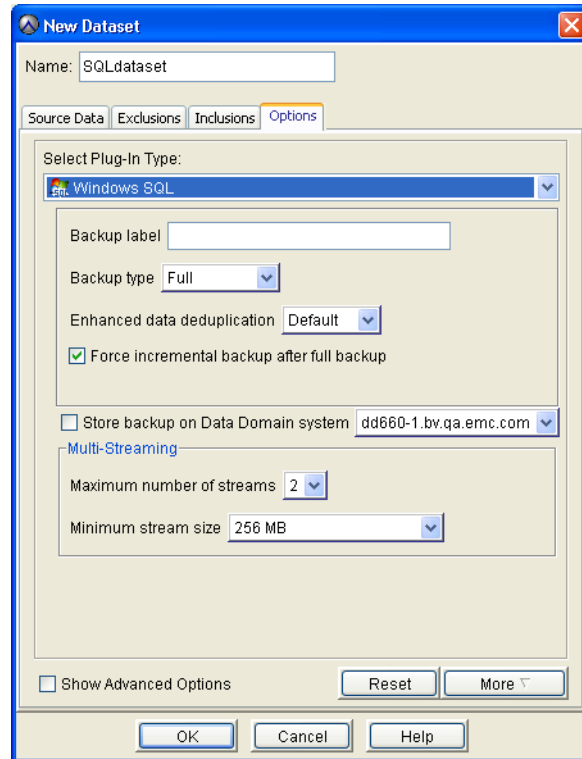
6. (Optional) Click the **Inclusions** tab and define databases or instances to include in the dataset that otherwise would have been excluded based on your selections on the **Exclusions** tab:
 - a. From the **Select Plug-In Type** list, select **Windows SQL**, as shown in the following figure.



- b. Click ... to browse to the instances or databases to include.
The Select Files And/Or Folders dialog box appears.
 - c. Select the client from the **Clients** tree in the left pane.
 - d. In the middle pane of the **Select Files And/Or Folders** dialog box, expand the **Windows SQL** plug-in node.
 - e. Under the **Windows SQL** plug-in node, select the database 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.
 - f. Select the data to include in the dataset that otherwise would have been 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.
 - g. Click **OK**.
The selected instances or databases are listed on the Inclusions tab.

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, as shown in the following figure.



- b. To include a descriptive label for the backups for this dataset, type the label in the **Backup label** box.
 - c. From the **Backup type** list, choose the type of backup:
 - Select **Full** to back up the entire database, including all objects, system tables, and data.
 - Select **Differential** to back up any data that has changed since the last full backup.
 - Select **Incremental** to back up only the transaction logs.
 - d. From the **Enhanced data deduplication** list, choose whether to use enhanced data deduplication, which typically reduces the amount of client data that must be sent to the server but requires additional client CPU resources:
 - To use the global enhanced data deduplication setting already set on the server, select **Default**.
 - To disable enhanced data deduplication, select **Disabled**.
 - To enable enhanced data deduplication, select **Enabled**.

- e. Choose whether to force a transaction log (incremental) backup that contains transactions that occur between full backups by selecting or clearing the **Force incremental backup after full backup** checkbox.

Forcing a transaction log backup between full backups ensures that a point-in-time recovery to a point in time between the two full backups can occur, if necessary.

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 in the advanced options to control how Avamar handles transaction log backups of databases that use the simple recovery model.

- f. To store the backups for this dataset on a Data Domain system instead of the Avamar server, select **Store backup on Data Domain system** and select the Data Domain system from the list.
- g. To specify the maximum number of streams to use for backup and restore, select a value from the **Maximum number of streams** list. [“Multi-streaming” on page 75](#) provides details on multi-streaming.
- h. To specify the minimum size of each data stream, select a value from the **Minimum stream size** list.

- i. To specify advanced backup options, select the **Show Advanced Options** checkbox. Otherwise, proceed to [step 8](#).

Additional options appear in red on the Options tab, as shown in the following figure.

The screenshot shows the 'New Dataset' dialog box with the 'Options' tab selected. The 'Name' field contains 'SQLdataset'. The 'Select Plug-In Type' dropdown is set to 'Windows SQL'. The 'Backup label' field is empty. The 'Backup type' dropdown is set to 'Full'. The 'Enhanced data deduplication' dropdown 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' and is highlighted in red. The 'For simple recovery model databases' dropdown is set to 'Skip incremental with error' and is highlighted in red. The 'Enable debugging messages' checkbox is unchecked. The 'Store backup on Data Domain system' checkbox is unchecked, and the dropdown is set to 'dd660-1.bv.qa.emc.com'. The 'Multi-Streaming' section has 'Maximum number of streams' set to 2 and 'Minimum stream size' set to 256 MB. The 'Authentication' section has 'SQL server address', 'SQL login ID', and 'SQL password' fields empty, and 'Authentication method' set to 'NT authentication'. The 'Show Advanced Options' checkbox is checked. At the bottom, there are 'Reset', 'More', 'OK', 'Cancel', and 'Help' buttons.

- j. Select or clear the **Force full backup** checkbox to specify whether to perform a full backup when Avamar detects a log gap or when there is no previous full backup, from which a transaction log (incremental) or differential backup can be applied, on the server for the database. Effectively, this option automates taking a full backup when necessary.

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 in the event that no existing full backup is present on the Avamar server.

- k. Select an option from the **Truncate database log** list to control database transaction log truncation behavior:
 - Select **(Default) Only for incremental backup** to truncate the log if the backup type is set to incremental (transaction log). No log truncation occurs if the backup type is full or differential.
 - Select **For all backup types** to truncate the log regardless of the backup type.

NOTICE

This setting breaks the chain of log backups and should not be used unless the backup type is set to incremental (transaction log).

- Select **Never** if you do not want to truncate the log under any circumstances.
- l. If you are performing an incremental (transaction log) backup and you have selected databases that use the simple recovery model, which does not support transaction log backups, then specify how Avamar handles the databases by selecting one of the following options from the **For simple recovery model databases** list:
 - **Skip incremental with error** — If you select databases that use the simple recovery model and databases that use other recovery models, then Avamar excludes the databases with the simple recovery model from the backup and writes an error message to the log. The backup completes with exceptions. If you select only databases that use the simple recovery model, then the backup fails.
 - **Skip incremental with warning** — If you select databases that use the simple recovery model and databases that use other recovery models, then Avamar excludes databases with the simple recovery model from the backup and writes a warning to the log for each database that uses the simple recovery model. The backup completes successfully. If you select only databases that use the simple recovery model, then the backup fails.
 - **Promote incremental to full** — Avamar performs a full backup instead of a transaction log backup for databases that use the simple recovery model.
 - m. Choose whether to write maximum information to log files by selecting or clearing the **Enable debugging messages** checkbox. If selected, very large log files are created.
 - n. (Optional) Type the hostname or IP address of the SQL server in the **SQL server address** box.
 - o. Choose whether to use NT authentication or SQL Server authentication to connect to SQL Server by selecting a value from the **Authentication method** list.
 - p. If you select the SQL Server authentication method, type the SQL Server login ID and password in the **SQL login ID** and **SQL password** boxes, respectively.

[Appendix A, “Plug-in Options,”](#) provides additional information on backup plug-in options.

8. Click **OK**.

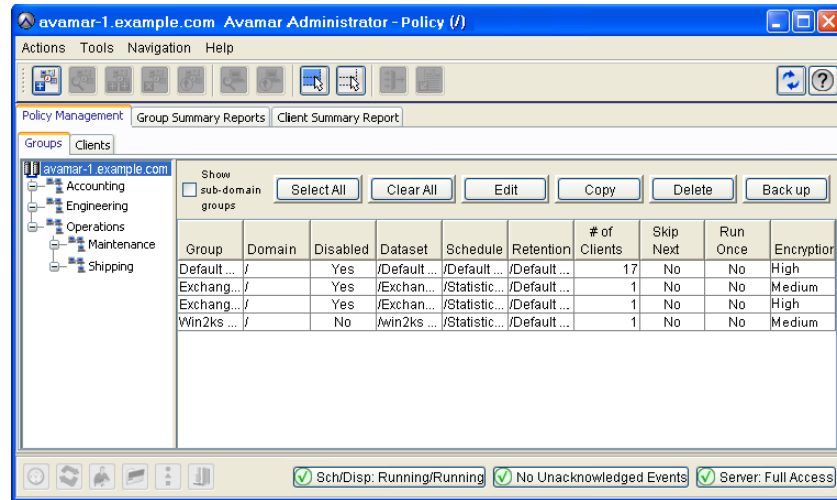
Creating a group



To create a group for scheduled SQL Server backups:

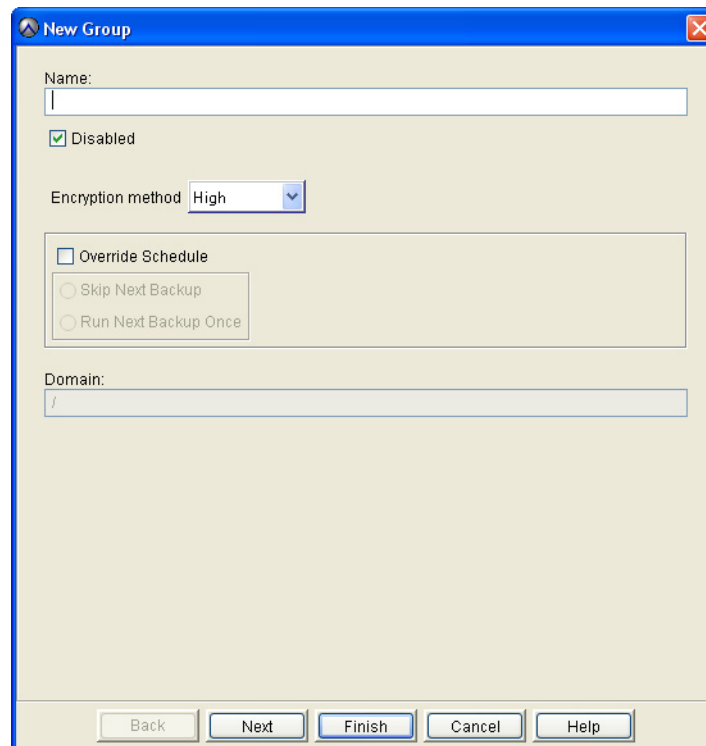
1. In Avamar Administrator, click the **Policy** launcher button.

The Policy window appears.



2. Select the **Groups** tab.
3. In the left pane, select the Avamar domain to which the group should belong.
4. Select **Actions > New Group**.

The New Group wizard appears.



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

Do not use any of the following characters in the group name:

~!@\$%^&(){}[]|,~;#\/*?<>'"&

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

Selecting the checkbox disables backups for the group.

- From the **Encryption method** list, select the encryption method for client/server network communications. This is the method that all clients in the group use unless the method is overridden at the client level.

The exact encryption technology and bit strength used for any given client-server connection depends on a number of factors, including the client platform and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

- Choose 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:

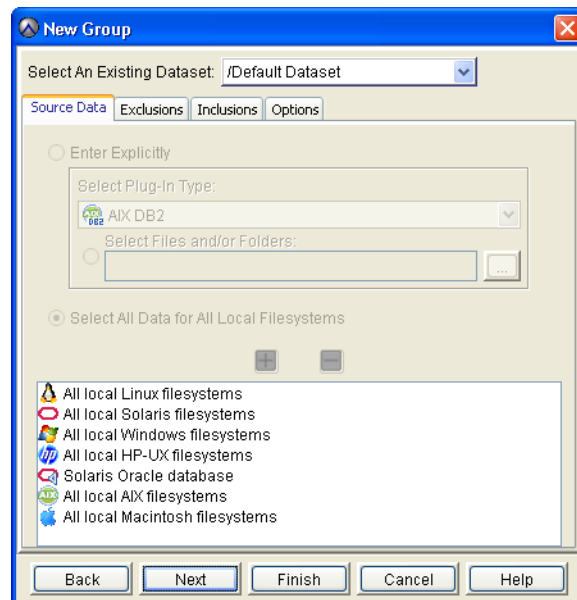
- Select **Override Schedule**.

Selecting Override Schedule enables the Skip Next Backup and Run Next Backup Once options.

- Choose whether to skip the next scheduled backup entirely or to perform the next scheduled backup one time only by selecting either **Skip Next Backup** or **Run Next Backup Once**.

- Click **Next**.

The next New Group wizard screen appears with dataset information.



- From the **Select An Existing Dataset** list, select the dataset that you defined during “Creating a dataset” on page 61.

11. Click **Next**.

The next New Group wizard screen appears with schedule information.

The screenshot shows the 'New Group' wizard window. At the top, there is a dropdown menu for 'Select An Existing Schedule:' with '/Default Schedule' selected. Below this, the 'Next Run Time:' is displayed as '2011-08-29 10:00 PM'. The 'Repeat this schedule' section has radio buttons for 'Daily', 'Weekly' (selected), 'Monthly', and 'On-Demand'. Underneath, checkboxes for days of the week are shown: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday, all of which are checked. There are 'Select All' and 'Unselect All' buttons. The 'Operating Time' section includes 'Earliest start time:' (10:00 PM), 'End no later than:' (08:00 AM), and 'Backup window duration:' (10 hours and 0 minutes). The 'Activation Constraints' section has 'Delay until:' (Thu 2011-07-28), 'No End Date' (selected), and 'End after:' (Mon 2011-08-29). At the bottom, there are 'Back', 'Next', 'Finish', 'Cancel', and 'Help' buttons.

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

You cannot edit schedules from this screen. Detailed schedule properties are shown so that you can review them prior to making a selection. The *EMC Avamar Administration Guide* provides additional information about editing schedule properties.

13. Click **Next**.

The next New Group wizard screen appears with retention policy information.

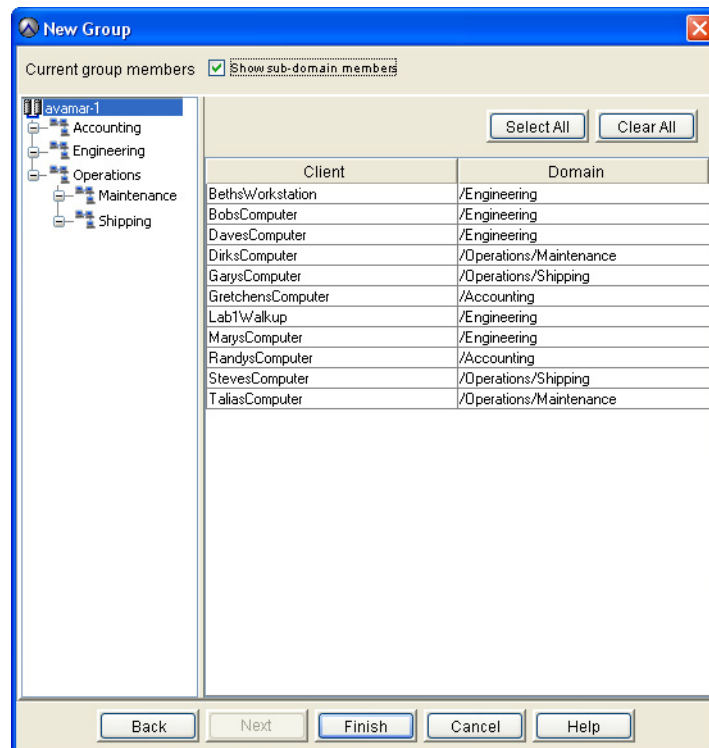
The screenshot shows the 'New Group' wizard window. At the top, there is a dropdown menu for 'Select An Existing Retention Policy:' with '/Default Retention' selected. The 'Basic Retention Policy' section has radio buttons for 'Retention period' (selected), 'End date', and 'No end date'. The 'Retention period' is set to '60' days. The 'End date' is set to 'Fri 2011-10-28'. There is a checkbox for 'Override basic retention policy for scheduled backups' which is checked, and an 'Advanced...' button next to it. At the bottom, there are 'Back', 'Next', 'Finish', 'Cancel', and 'Help' buttons.

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

You cannot edit retention policies from this screen. Detailed retention policy properties are shown so that you can review them prior to making a selection. The *EMC Avamar Administration Guide* provides additional information about editing retention policy properties.

15. Click **Next**.

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



16. Select the SQL server client from the client list.

If you are backing up databases in a SQL Server 2012 AlwaysOn availability group, then select the hostname of the physical node that is hosting the primary replica. [“Backup requirements for AlwaysOn availability groups” on page 38](#) explains how to determine which node is hosting the primary replica.

If you do not know which node is hosting the primary replica at the time of the backup, you can select the hostnames for all physical nodes that might host the availability group. The backup succeeds on the node that is hosting the primary replica at the time of the backup and fails on the nodes that are hosting the secondary replicas.

If SQL Server is configured as a Failover Cluster Instance (FCI) and you are backing up databases on shared storage in the cluster, then select the Avamar Cluster Client.

17. Click **Finish**.

The New Group wizard closes and the new group appears in the Policy window.

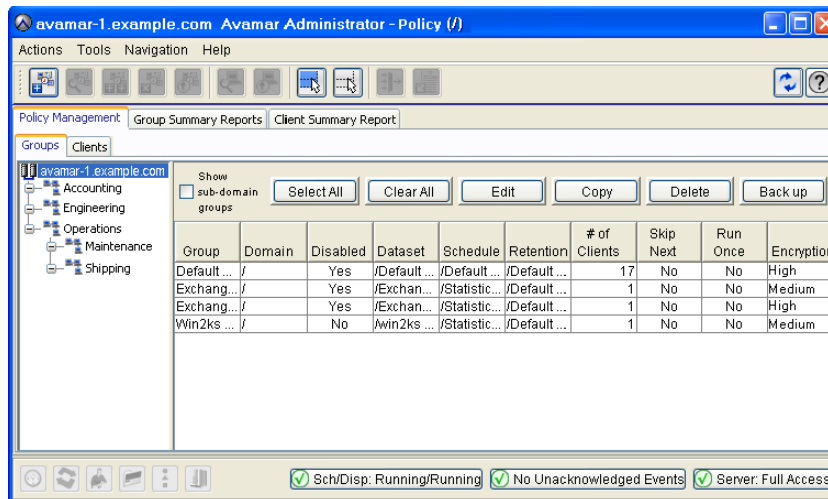
Enabling scheduled backups

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

1. In Avamar Administrator, click the **Policy** launcher button.



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 71.
4. Select **Actions > Group > Disable Group**.

This clears the **Disable Group** option on the **Actions > Group** menu.

A confirmation message appears.

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

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. The number of streams that you use to back up a database is automatically used to restore the database.

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 six streams for each backup. [“Specifying the maximum number of streams”](#) on page 78 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 choose to 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 79 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 115](#) 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 an SQL server with significant system resources, such as:

- ◆ Fast, multi-processor 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 at the same time 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 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 choose to 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 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 you set the minimum stream size to the default setting of 256 MB, then the number of streams that are used to perform 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 number of streams that you configured for the backup, then the backup of the database uses only the maximum number of streams.

Understanding the order of databases in a backup

When multiple databases are included in a backup, Avamar sets the order in which the databases are backed up 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 maximum number of streams in the backup plug-in options, then Avamar can use those streams to back up another database while the first database is backed up. 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 six streams for each backup.

Determining the optimal value for the maximum number of streams

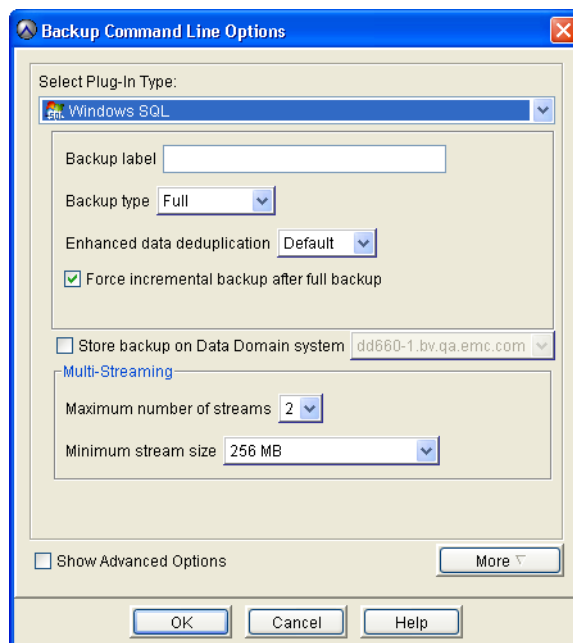
In environments with significant system resources, as described in [“When to use multi-streaming” on page 76](#), specify the maximum of six streams to achieve the best possible performance.

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.

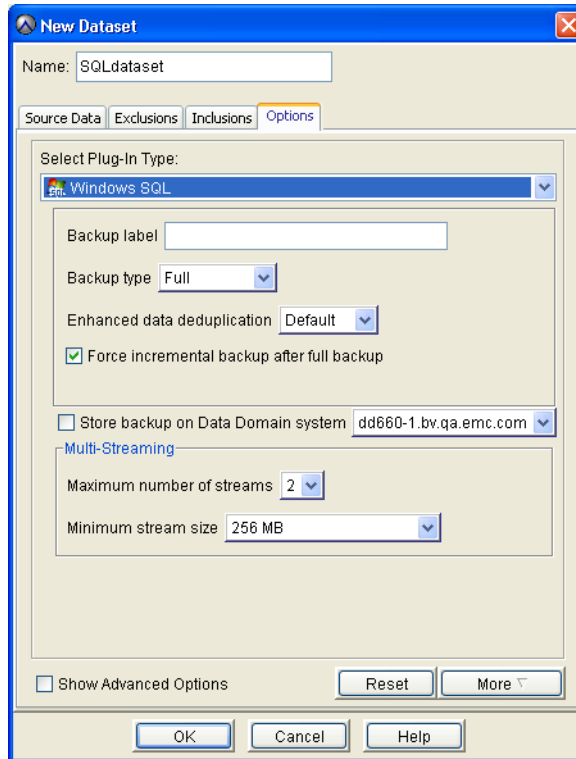
The optimal number of streams depends on the environment. In general, start with the highest maximum number of streams that you believe the environment can support. If resource usage during a backup is excessive and adversely impacts other processes on the server, then consider reducing the maximum number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

How to specify the maximum number of streams

During an on-demand backup, you select the maximum number of streams using the Maximum number of streams list on the Backup Command Line Options dialog box.



For scheduled backups, you select the maximum number of streams on the Options tab when you create a dataset.



You cannot specify the maximum number of streams for a restore. The number of streams that you use to back up the data is also used to restore the data.

Specifying the minimum stream size

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.

Determining the optimal value for the minimum stream size

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 76](#) and [“Understanding the order of databases in a backup” on page 77](#) 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 a set of 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 5 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	Select any of the following values: <ul style="list-style-type: none"> • 256 MB (default) • 512 MB • 1,024 MB • 2,048 MB 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.

Table 5 Recommended minimum stream size by backup set and environment (page 2 of 2)

Backup set and environment	Recommended minimum stream size
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 at the same time.</p>
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 82 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 82 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 6 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	6	256 MB	Avamar uses all six streams to back up the 9 GB database. When the backup of the 9 GB database completes, Avamar uses all six streams to back up the 100 MB databases, with one stream for each database.
	6	2,048 MB	Avamar uses four streams to back up the 9 GB database, and the remaining two to back up two of the 100 MB databases. When the backup of the 9 GB database completes, Avamar uses all six 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	6	2,048 MB	Avamar uses all six 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 six streams to back up the small databases, with one stream for each database, in order from the largest database to the smallest database.

Table 6 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.

How to specify 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. [“How to specify the maximum number of streams” on page 78](#) provides illustrations of these dialog boxes.

CHAPTER 4

Restore

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

- ◆ Restore requirements 86
- ◆ Finding a backup..... 88
- ◆ Restoring to the original location..... 94
- ◆ Restoring to a new database in the original instance 98
- ◆ Restoring to a different instance 101
- ◆ Restoring to a file 107
- ◆ Restoring system databases..... 121
- ◆ Restoring a database in an AlwaysOn availability group 127
- ◆ Restoring a database with an intact log file 128
- ◆ Setting restore options..... 129

Restore requirements

Review the following system requirements before you perform a restore:

- ◆ [“Software requirements for restore” on page 86](#)
- ◆ [“Tail-log backup and point-in-time restore requirements” on page 86](#)
- ◆ [“Requirements to restore a database in an AlwaysOn availability group” on page 87](#)
- ◆ [“Requirements to restore secondary database files” on page 87](#)
- ◆ [“Requirements to restore the Report Server database” on page 87](#)
- ◆ [“SQL Server write permission requirements” on page 87](#)

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.

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 the Tail-log backup option is selected.

Requirements to restore a database in an AlwaysOn availability group

When you restore a database in an AlwaysOn availability group on either the primary replica or a secondary replica, remove the database from the availability group. Details on the restore procedures are available in [“Restoring a database in an AlwaysOn availability group” on page 127](#).

Requirements to restore secondary database files

If you are restoring the secondary 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 files after you have already performed a restore with a tail-log backup.

If you need to restore the secondary files of a database with two different restore operations, then you must 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 properly.

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 to restore either by date or by files and folders (or specific instance or database, in the case of SQL Server). The following topics provide details on finding a backup for a restore:

- ◆ [“When to find a backup by date” on page 88](#)
- ◆ [“How to find a backup by date” on page 88](#)
- ◆ [“When to find a backup by file or folder” on page 91](#)
- ◆ [“How to find a backup by file or folder” on page 91](#)

When to find a backup by date

Locate backups by date when:

- ◆ All instances and databases for a client are backed up in a single backup set.
- ◆ 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 disaster recovery backups every Wednesday and Saturday night, and you run full volume backups daily. If you need to rebuild a server, you can select the disaster recovery backup with the date closest to the event that caused the loss of data.

How to find a backup by date



To find backups for a restore by date:

1. In Avamar Administrator, click the **Backup & Restore** launcher button.

The Backup and Restore window appears.
2. In the clients tree, select the SQL server. Keep the following points in mind when you select a client:
 - If you are restoring databases in a SQL Server 2012 AlwaysOn availability group, then select the physical node that is hosting the primary replica. [“Backup requirements for AlwaysOn availability groups” on page 38](#) explains how to determine which node is hosting the primary replica.
 - If SQL Server is configured as a Failover Cluster Instance (FCI) and you are restoring a database on shared storage in the cluster, then select the Avamar Cluster Client.
 - You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
3. Click the **Select for Restore** tab.
4. Click the **By Date** tab.
5. Select a backup from the calendar:
 - a. Use the year and month navigational arrows to browse to a backup.

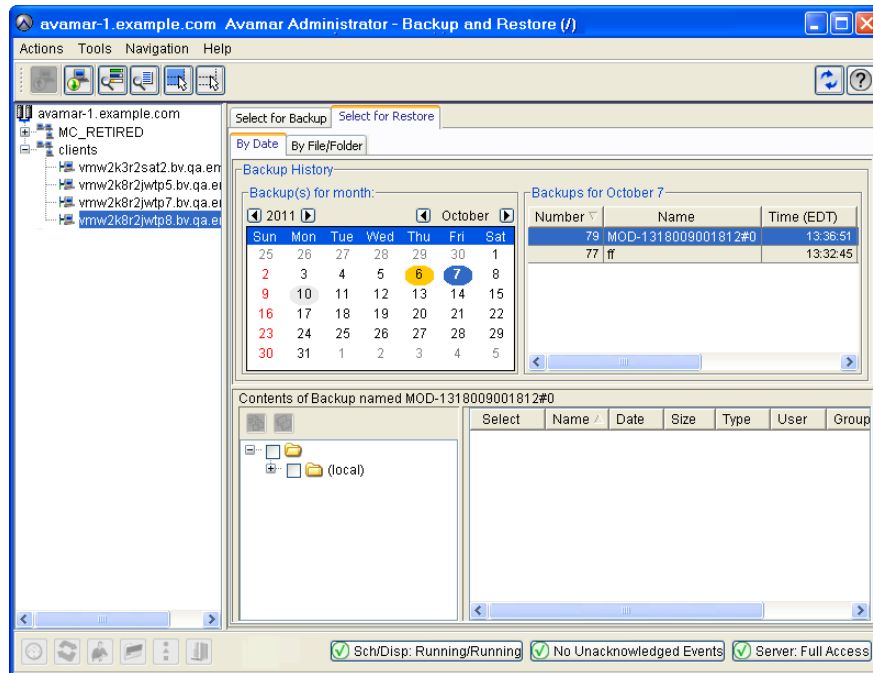
Dates highlighted by yellow indicate a valid backup was performed on that date.

b. Click a date highlighted by yellow.

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

If a backup on the selected date is the second or later full backup and the Force incremental backup after full backup checkbox was selected for the backup, then three backups are listed in the Backups table:

- The first backup in the list displays 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 displays 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 includes only the full backup.

6. 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 bottom 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.
7. In the two bottom panes of the **Backup and Restore** 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 bottom left pane.
 - To restore a database and its logs, expand the node for the instance in the folder tree in the bottom 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 bottom left pane, select the database in the bottom left pane, and then select the checkbox next to the files in the filegroup in the bottom 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 bottom 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.

8. Continue with the restore as described in the following topics:
 - [“Restoring to the original location” on page 94](#)
 - [“Restoring to a new database in the original instance” on page 98](#)
 - [“Restoring to a different instance” on page 101](#)
 - [“Restoring to a file” on page 107](#)
 - [“Restoring system databases” on page 121](#)

When to find a backup by file or folder

Locate backups by the specific files or folders contained within each 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 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 file.
- ◆ The date of the backup or what was saved in a backup is unknown, but you know the name of the database or instance.

How to find a backup by file or folder

To find a backup by specific files or folders (or instances or databases, in the case of SQL Server) in that backup:



1. In Avamar Administrator, click the **Backup & Restore** launcher button.

The Backup and Restore window appears.
2. In the clients tree, select the SQL server. Keep the following points in mind when you select a client:
 - If you are restoring databases in a SQL Server 2012 AlwaysOn availability group, then select the physical node that is hosting the primary replica. [“Backup requirements for AlwaysOn availability groups” on page 38](#) explains how to determine which node is hosting the primary replica.
 - If SQL Server is configured as a Failover Cluster Instance (FCI) and you are restoring a database to shared storage in the cluster, then select the Avamar Cluster Client.
 - You cannot view clients outside the domain for the login account. To view all clients, log in to the root domain.
3. Click the **Select for Restore** tab.
4. Click the **By File/Folder** tab.

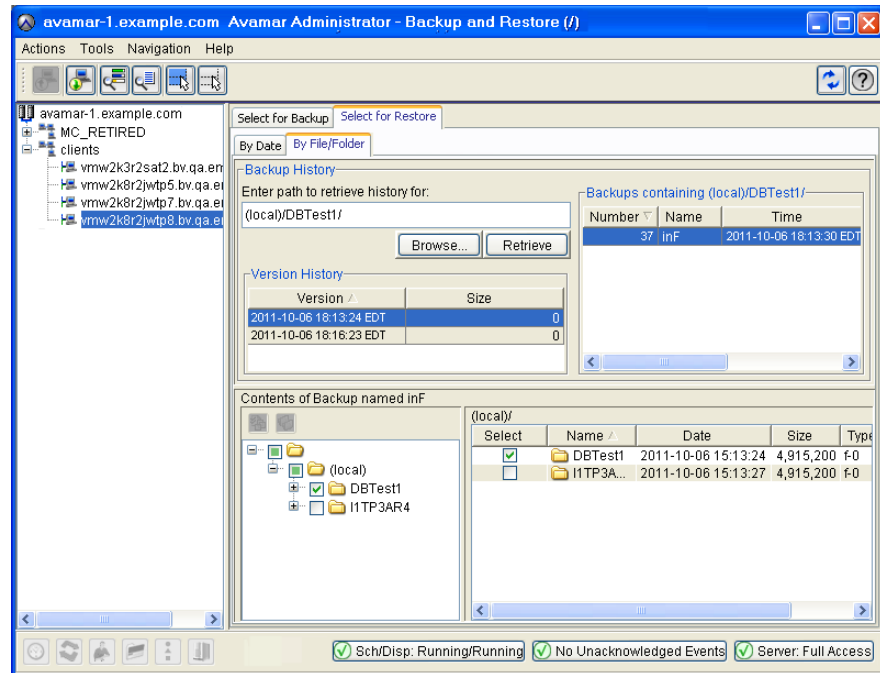
5. In the **Enter path to retrieve history for** text box, specify the path to the instance or database by using one of the following methods:
 - Browse to the instance or database:
 - a. Click **Browse**.

The **Select File or Folder** dialog box appears.
 - b. Select the client from the **Clients** tree in the left pane.
 - c. In the middle pane, expand the **Windows SQL** plug-in node.
 - d. Under the **Windows SQL** plug-in node, select the database 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.
 - e. 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.
 - f. Click **OK**.
 - Type the path to the instance or database using one of the following formats:
 - To restore the local instance, type (local).
 - To restore a database in the local instance, type (local)/DATABASE/.
 - To restore a named instance, type CLIENT\INSTANCE/.
 - To restore a database in a named instance, type CLIENT\INSTANCE/DATABASE/.
 - To restore a database if there is only one instance on the client and it is not named local, type CLIENT/DATABASE/.where CLIENT is the name of the SQL server, INSTANCE is the name of the named instance, and DATABASE is the name of the database.
6. Click **Retrieve**.

The Version History table lists all versions and sizes for that directory or file that have been backed up from the selected client.
7. Select the directory or file 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.

8. 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 bottom 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.

9. In the two bottom panes of the **Backup and Restore** 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 bottom left pane.
- To restore a database and its logs, expand the node for the instance in the folder tree in the bottom 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 bottom left pane, select the database in the bottom left pane, and then select the checkbox next to the files in the filegroup in the bottom 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 bottom 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.

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

- [“Restoring to the original location” on page 94](#)
- [“Restoring to a new database in the original instance” on page 98](#)
- [“Restoring to a different instance” on page 101](#)
- [“Restoring to a file” on page 107](#)
- [“Restoring system databases” on page 121](#)

Restoring to the original location

The following topics provide details on restoring a SQL Server instance, database, or filegroup to its original location:

- ◆ [“Understanding the options when restoring to the original location” on page 94](#)
- ◆ [“How to restore to the original location” on page 95](#)

Understanding the options when restoring to the original location

When you restore a SQL Server instance, database, or filegroup to its original location, you can either perform a standard restore with a tail-log backup and recovery, or you can 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 have not been included 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 required, 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, which overrides a SQL Server safety check that is intended to prevent you from accidentally overwriting a different database or file. This safety check is described in the Microsoft Transact-SQL Reference Manual under the RESTORE command section.

How to restore to the original location

To create a tail-log backup and perform a direct restore of a SQL Server instance, database, filegroup, or file with recovery:

1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 86](#).



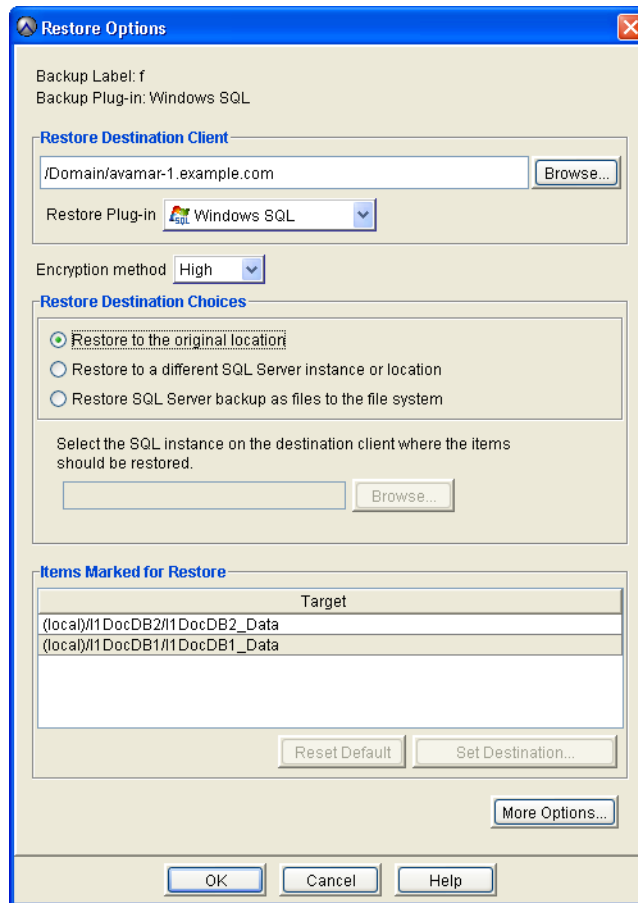
2. In Avamar Administrator, click the **Backup & Restore** launcher button.

The Backup and Restore 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 88](#)
 - [“How to find a backup by file or folder” on page 91](#)

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

The Restore Options dialog box appears.



5. Ensure that the **Windows SQL** plug-in is selected in the **Restore Plug-in** list.

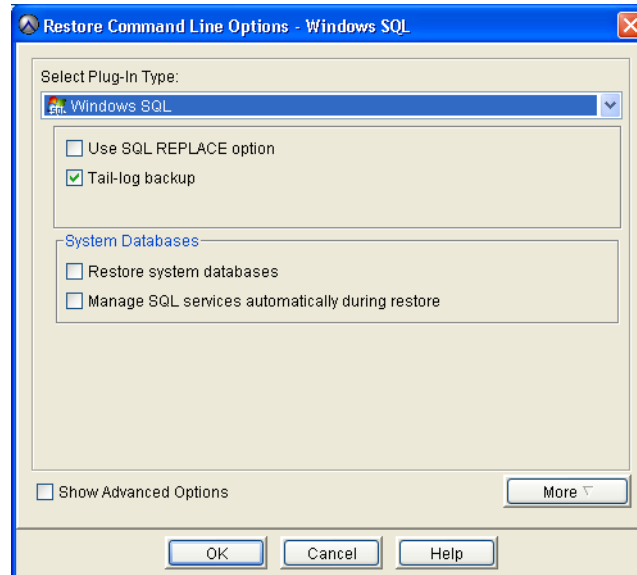
6. Select the encryption method to use for client/server data transfer during the restore.

The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform 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. To restore system databases, set the plug-in options as discussed in [“System database restore options”](#) on page 132.

11. (Optional) Select **Show Advanced Options**, and set the advanced options as discussed in the following topics:

- [“Recovery operation options”](#) on page 131
- [“Authentication options”](#) on page 134
- [“Point-in-time recovery options”](#) on page 134

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

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

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

The Restore Request dialog box indicates that the restore was initiated.

14. 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. 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, and any transactions in the tail-log may be lost.

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 86](#).

2. In Avamar Administrator, click the **Backup & Restore** launcher button.

The Backup and Restore window appears.

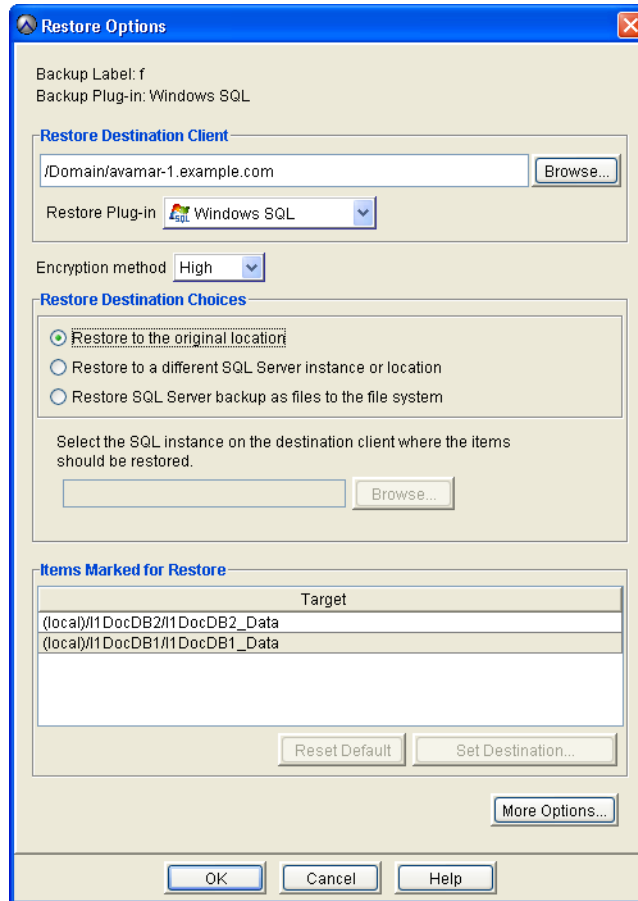
3. Find the backup and select the database to restore, as discussed in the following topics:

- [“How to find a backup by date” on page 88](#)
- [“How to find a backup by file or folder” on page 91](#)



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

The Restore Options dialog box appears.



5. Ensure that the **Windows SQL** plug-in is selected in the **Restore Plug-in** list.

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

The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform 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.

10. Leave the **Use SQL REPLACE option** checkbox clear.
11. Choose whether to perform a tail-log backup during the restore by selecting or clearing the **Tail-log backup** checkbox. [“Tail-log backup” on page 130](#) provides details.
12. In the **New database name** box, specify the new database name.
13. 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.

14. Choose 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.
15. (Optional) Set other plug-in options as discussed in the following topics:
 - [“Recovery operation options” on page 131](#)
 - [“Authentication options” on page 134](#)
 - [“Point-in-time recovery options” on page 134](#)

Disregard the system databases options, which are only necessary when you are restoring a system database.
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 was initiated.
18. 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. 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, and any transactions in the tail-log may be lost.

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:

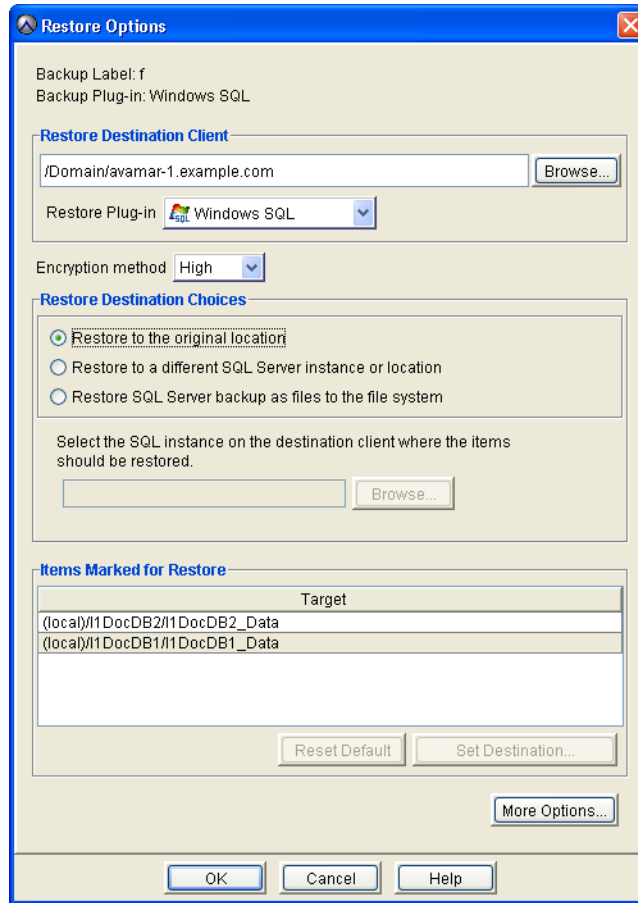


1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 86](#).
2. In Avamar Administrator, click the **Backup & Restore** launcher button.

The Backup and Restore 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 88](#)
 - [“How to find a backup by file or folder” on page 91](#)

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

The Restore Options dialog box appears.



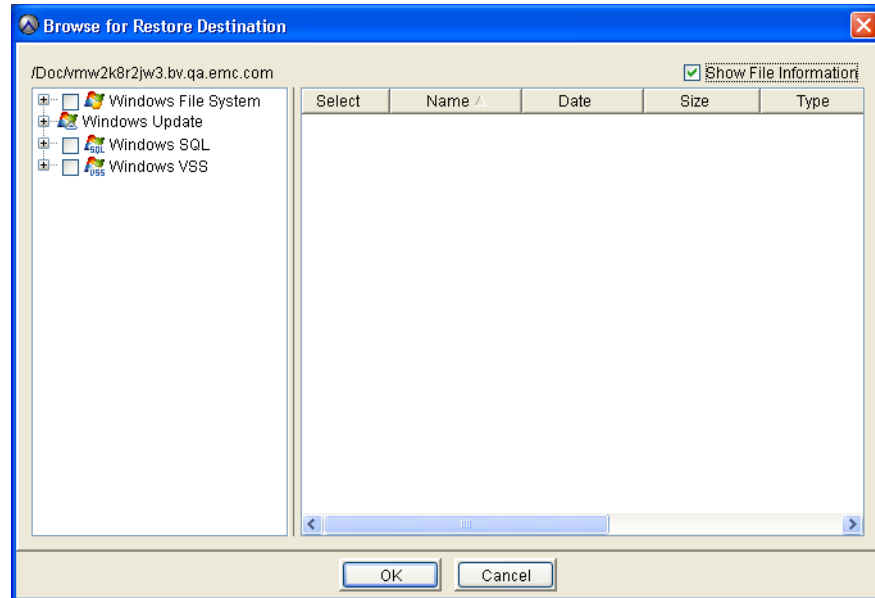
5. Ensure that the **Windows SQL** plug-in is selected in the **Restore Plug-in** list.
6. Select the encryption method to use for client/server data transfer during the restore.

The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform 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.

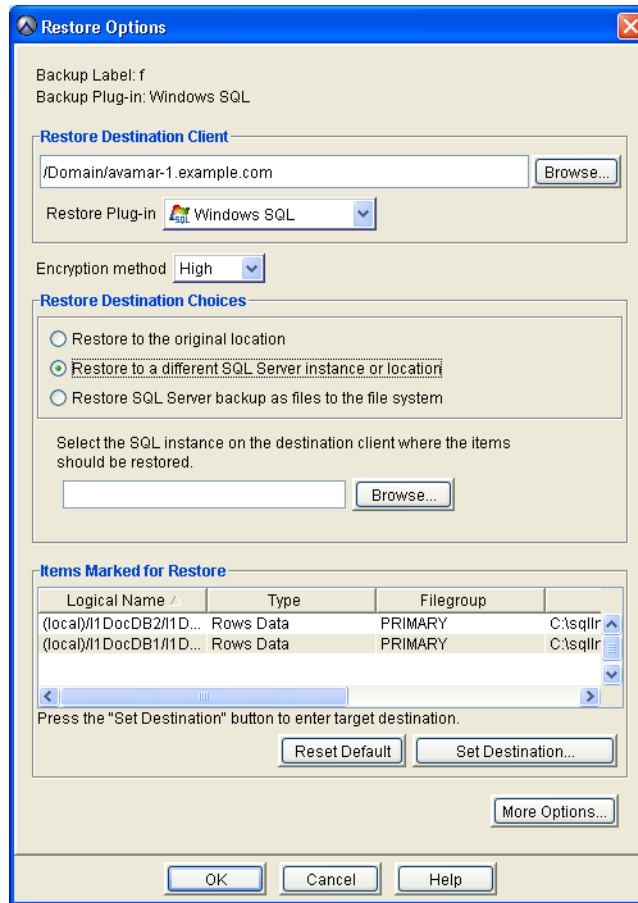
8. Select the instance on the destination client for the restore:
 - a. Next to the **Select the SQL instance on the destination client where the items should be restored** box, click **Browse**.

The Browse for Restore Destination dialog box appears.



- b. Select the **Windows SQL** node in the left pane.
 - c. In the right pane, select the checkbox next to the instance.
 - d. Click **OK**.

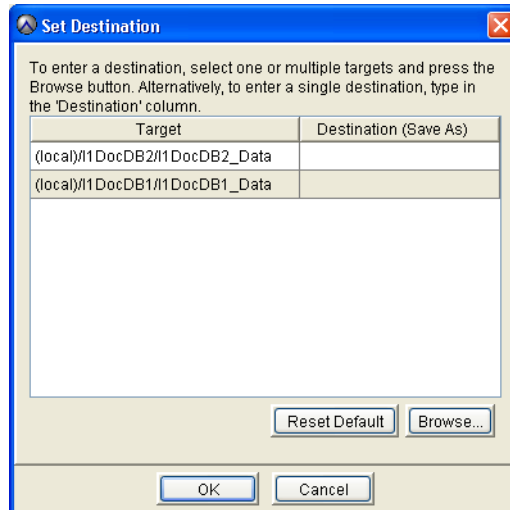
9. Select **Restore to a different SQL Server instance or location**.



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

- a. Click **Set Destination** below the **Items Marked for Restore** list.

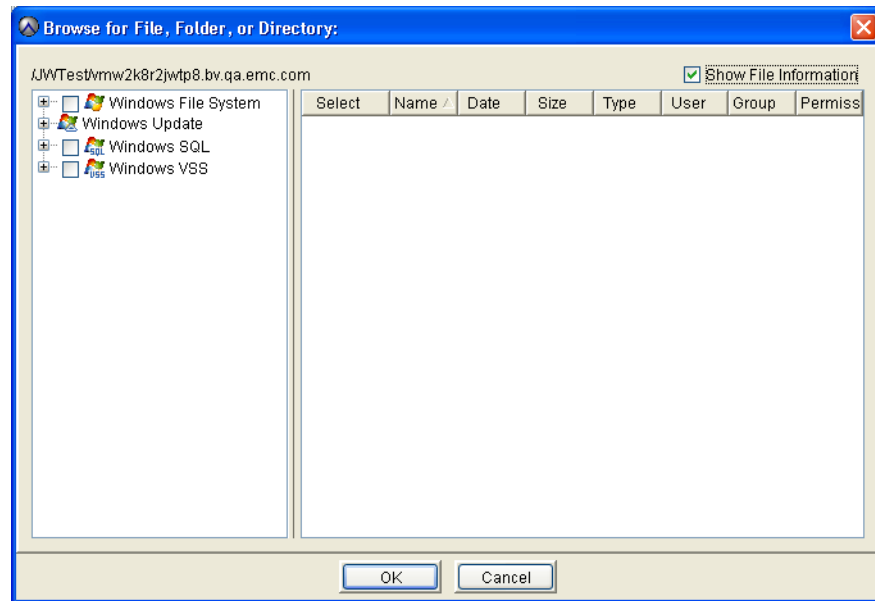
The Set Destination dialog box appears.



- 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 the **Windows File System** node in the left pane.
- e. In the right pane, browse to and select the checkbox for the directory 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
- 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, there is a 'More' button and '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, which is not supported when you restore to a different SQL Server instance.
15. To restore system databases, set the plug-in options as discussed in [“System database restore options”](#) on page 132.
16. (Optional) To restore the database with a new name, type the new name in the **New database name** box.
17. Disregard the remaining **Redirected Restore** settings (**Alternate database location**, **Alternate log location**, and **Path to alternate log location**), since you already specified these settings on the **Set Destination** dialog box.

18. (Optional) Set other plug-in options as discussed in the following topics:

- [“Recovery operation options” on page 131](#)
- [“Authentication options” on page 134](#)
- [“Point-in-time recovery options” on page 134](#)

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

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

The Restore Request dialog box indicates that the restore was initiated.

21. 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 are not provided by the Avamar Plug-in for SQL Server, 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. The following topics provide details:

- ◆ [“Restoring to a file with the SQL Server plug-in” on page 107](#)
- ◆ [“Restoring to a file with the Windows file system plug-in” on page 112](#)
- ◆ [“Restoring a database with SQL Server tools” on page 115](#)

Restoring to a file with the SQL Server plug-in

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

1. Ensure that the environment meets the guidelines in [“Restore requirements” on page 86](#).

2. In Avamar Administrator, click the **Backup & Restore** launcher button.



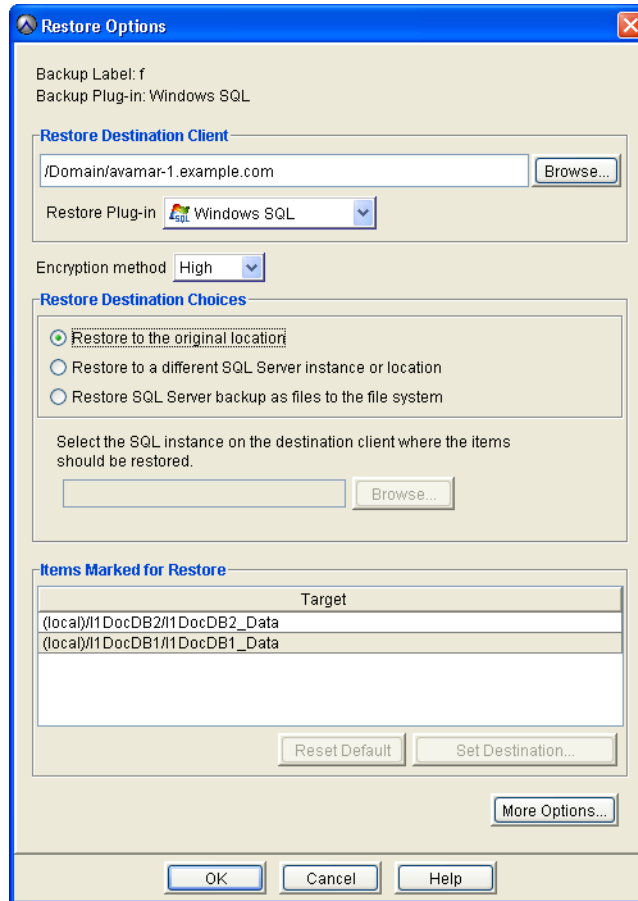
The Backup and Restore window appears.

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 88](#)
- [“How to find a backup by file or folder” on page 91](#)

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

The Restore Options dialog box appears.

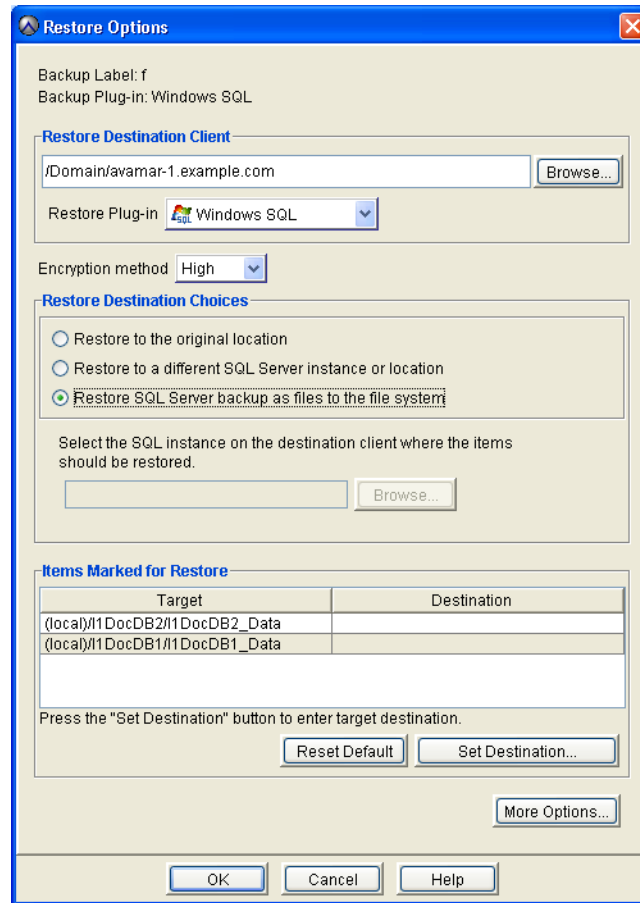


5. Ensure that the **Windows SQL** plug-in is selected in the **Restore Plug-in** list.
6. Select the encryption method to use for client/server data transfer during the restore.

The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

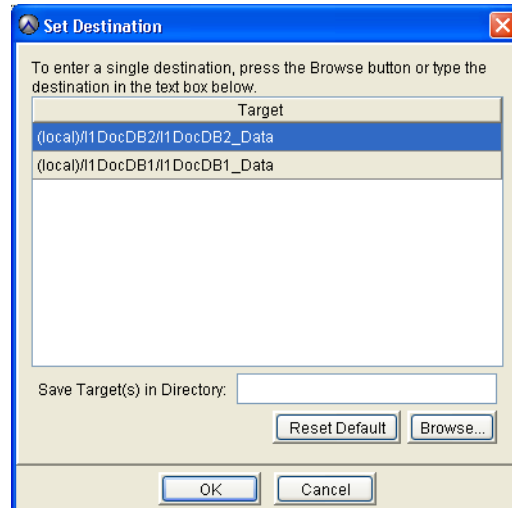
7. Specify the destination client in the **Restore Destination Client** box:
 - To restore to the original client, leave the default setting of the original client domain and name. (You will select a different instance in the next step.)
 - To restore to a different client, click **Browse** and then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

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



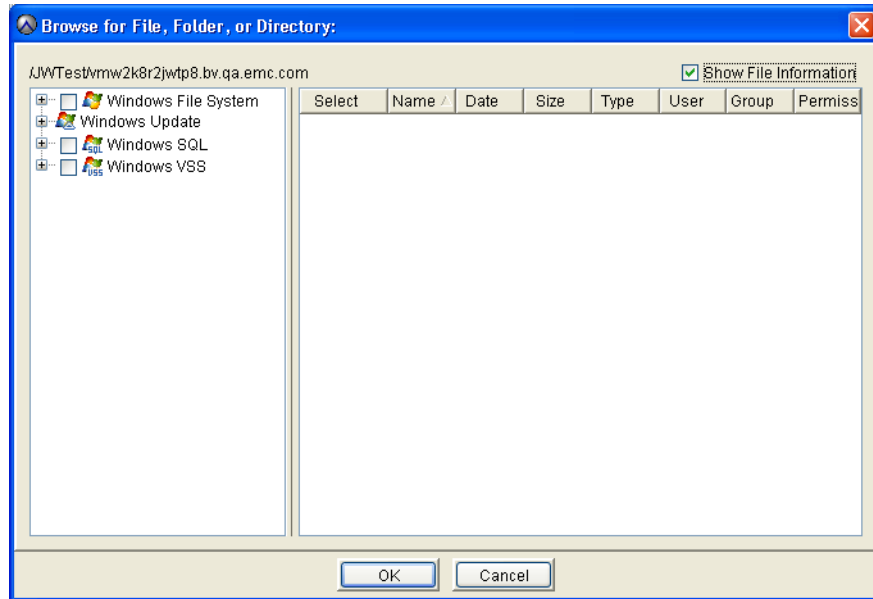
9. Set the destination file path for the database and log files to restore:
- Click **Set Destination** below the **Items Marked for Restore** list.

The Set Destination dialog box appears.



- b. Click **Browse**.

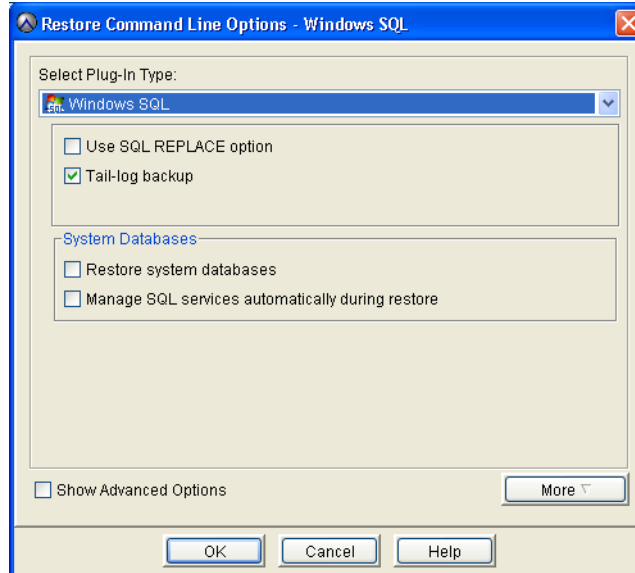
The Browse for File, Folder, or Directory dialog box appears.



- c. Select the **Windows File System** node in the left pane.
- d. In the right pane, browse to and select the checkbox for the directory 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 system databases are included in the restore, 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 are restoring to a file.
 - c. Click **OK**.
11. Click **OK**.

The Restore Request dialog box indicates that the restore was initiated.

12. Click **Close**.

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

DESTINATION \ INSTANCE \ DATABASE \ FILENAME

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.
- FILENAME is the name of the file. There may be multiple files for a single backup, depending on the number of streams that were used to perform 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 using SQL Server tools, as described in [“Restoring a database with SQL Server tools” on page 115](#).

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 86](#).



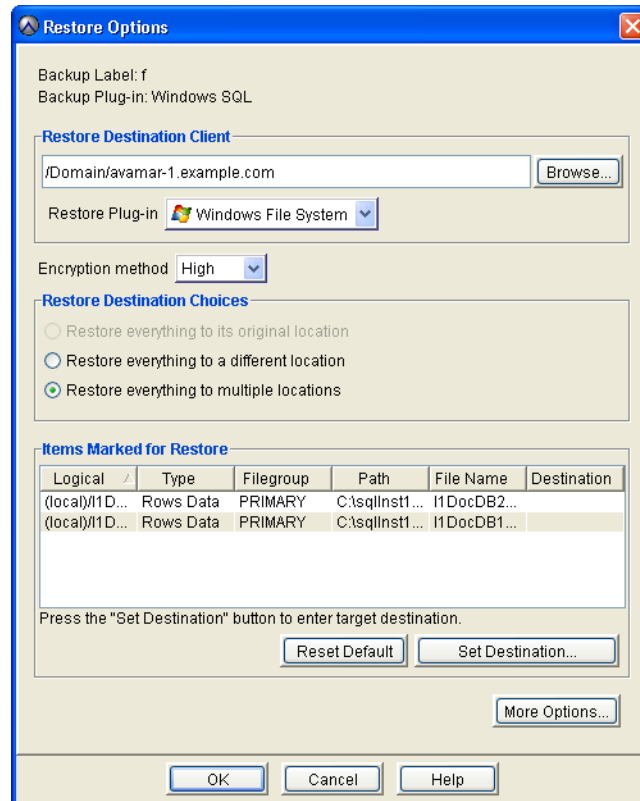
2. In Avamar Administrator, click the **Backup & Restore** launcher button.

The Backup and Restore window appears.

3. Find the backup and select the database to restore:
 - [“How to find a backup by date” on page 88](#)
 - [“How to find a backup by file or folder” on page 91](#)
4. Select **Actions > Restore Now**.

The Restore Options dialog box appears.

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



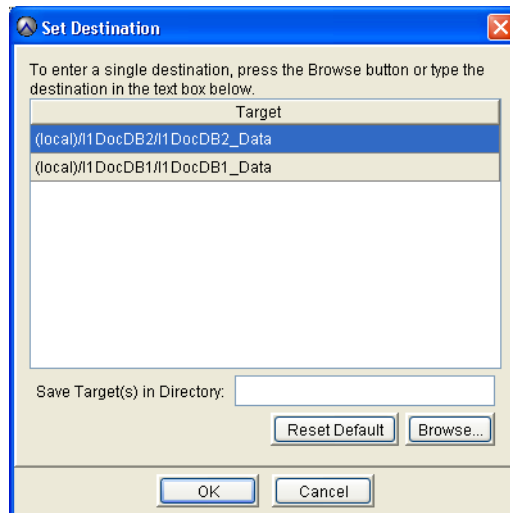
6. Select the encryption method to use for client/server data transfer during the restore.

The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform 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**.

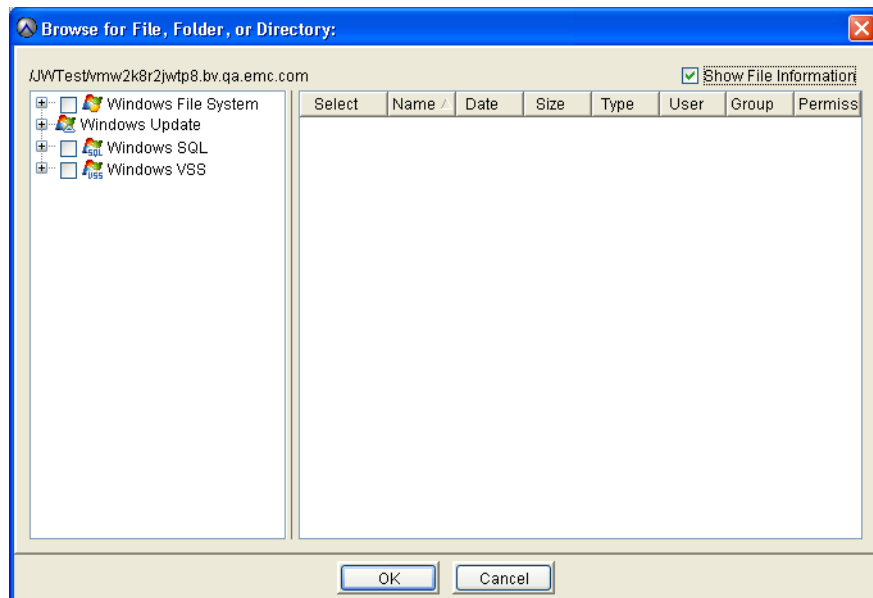
9. Set the destination file path for the database and log files to restore:
 - a. Click **Set Destination** below the **Items Marked for Restore** list.

The Set Destination dialog box appears.



- b. Click **Browse**.

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



- c. Select the **Windows File System** node in the left pane.
 - d. In the right pane, browse to and select the checkbox for the directory 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 was initiated.

12. Click **Close**.

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

`DESTINATION\DATABASE\FILENAME`

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.
- `FILENAME` is the name of the file. There may be multiple files for a single backup, depending on the number of streams that were used to perform 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 using SQL Server tools, as described in [“Restoring a database with SQL Server tools” on page 115](#).

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, as described in the following topics:

- ◆ [“Restoring a database with SQL Server Management Studio” on page 115](#)
- ◆ [“Restoring a database with a Transact-SQL RESTORE command” on page 118](#)
- ◆ [“Restoring a database with the sqlcmd utility” on page 119](#)

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 as described in either “[Restoring to a file with the SQL Server plug-in](#)” on page 107 or “[Restoring to a file with the Windows file system plug-in](#)” on page 112.
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 initiate 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 initiate 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, as well as 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 107](#) or [“Restoring to a file with the Windows file system plug-in” on page 112](#).
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:

```
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 should remain in the restoring state, which allows 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 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 appropriate 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:

```
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 using 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 Microsoft SQL Server documentation provides more information about using 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 SQLSERVER -E
1> restore database DBNAME
2> from disk = 'DRIVE:\FOLDER\f-0.stream0'
3> with recovery;
4> go
```

where:

- ◆ SQL Server is the 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 that is being restored:

```
sqlcmd -S SQLSERVER -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 'C:\DBNAME.mdf',
5> move 'MyDatabase_log' to 'C:\DBNAME.ldf'
6> go
```

where MyDatabase is the database to restore and DATA-FILE is the path and file name of the backup file from which to restore the database.

To restore a database to a point in time, use STOPAT syntax, as shown in the following example:

```
RESTORE DATABASE MyDatabase
FROM disk= 'DATA-FILE'
WITH NORECOVERY, STOPAT = 'Apr 25, 2013 12:00 AM'
RESTORE LOG MyDatabase
FROM disk= 'LOG-FILE'
WITH RECOVERY, STOPAT = 'Apr 25, 2013 12:00 AM'
```

where MyDatabase is the database to restore, DATA-FILE is the path and file name of the full backup file from which to restore the database, and LOG-FILE 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. However, the restore might be required if one or more system databases are damaged.

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. The following topics provide instructions for both methods:

- ◆ [“Restoring system databases automatically” on page 121](#)
- ◆ [“Restoring system databases manually” on page 124](#)

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 86](#).
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.

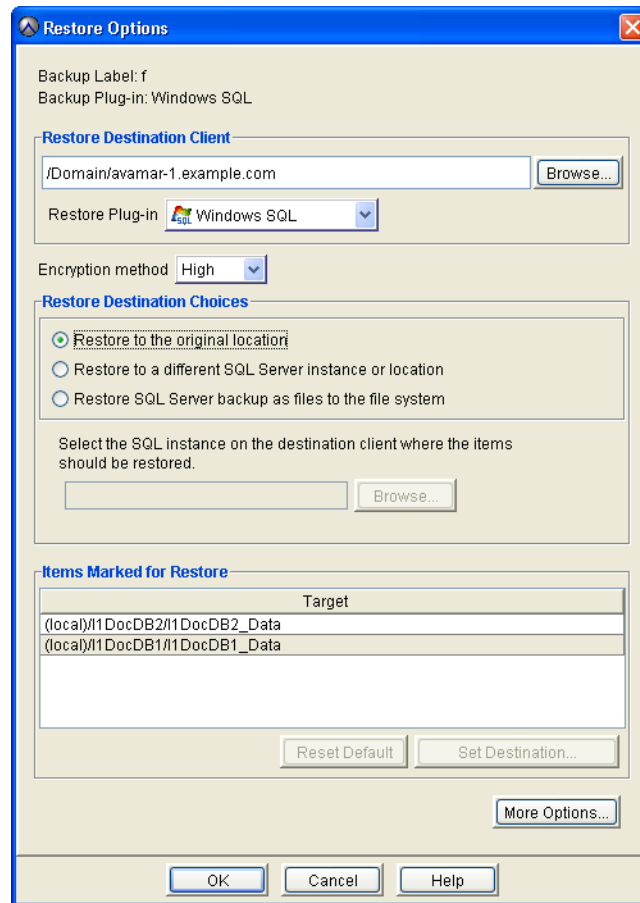
The Backup and Restore window appears.

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 88](#)
 - [“How to find a backup by file or folder” on page 91](#)



5. Select **Actions > Restore Now**.

The Restore Options dialog box appears.

6. Ensure that the **Windows SQL** plug-in is selected in the **Restore Plug-in** list.

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

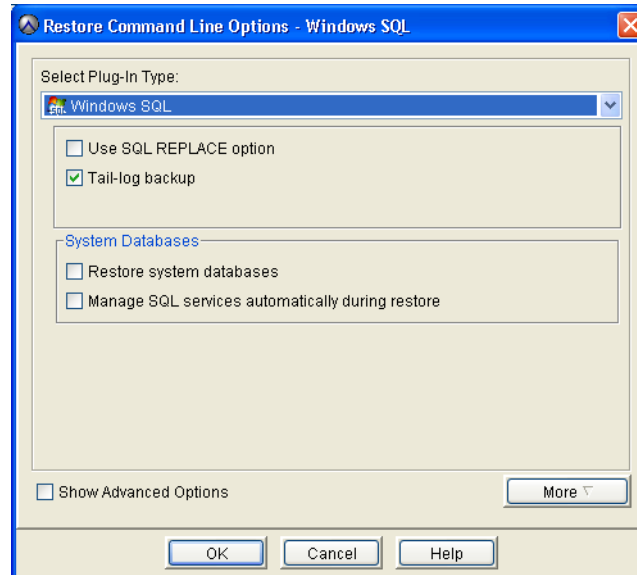
The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

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 107](#) 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 131](#)
- [“Authentication options” on page 134](#)

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.

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

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

The Restore Request dialog box indicates that the restore was initiated.

17. 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, like the SQL Server Agent service and the Analysis Service, are stopped.
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:


```
cd \MSSQLPATH\Binn
sqlservr.exe -m -c
```
 - To start a named instance in single-user mode, open a command prompt and type:


```
cd \MSSQLPATH\Binn
sqlservr.exe INSTANCE -m -c
```

where \MSSQLPATH\Binn is the path to the Binn directory 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 properly 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 proceed to [step 5](#).

Complete the following steps if SQL services do not start in single-user mode properly:

- 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 its original location:
 - a. Ensure that the environment meets the guidelines in [“Restore requirements” on page 86](#).
 - b. In Avamar Administrator, click the **Backup & Restore** launcher button.
The Backup and Restore 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 88](#)
 - [“How to find a backup by file or folder” on page 91](#)
 - d. Select **Actions > Restore Now**.
The Restore Options dialog box appears.
 - e. Ensure that the **Windows SQL** plug-in is selected in the **Restore Plug-in** list.
 - f. Select the encryption method to use for client/server data transfer during the restore.

The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform 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 134](#).
 - l. Disregard the remaining restore options, which do not apply when you are restoring 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 was initiated.
 - 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:
`net start MSSQLServer`
 - To start a named instance of SQL Server, open a command prompt and type:
`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 86](#).
 - b. In Avamar Administrator, click the **Backup & Restore** launcher button.
The Backup and Restore 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 88](#)
 - [“How to find a backup by file or folder” on page 91](#)
 - d. Select **Actions > Restore Now**.
The Restore Options dialog box appears.
 - e. Ensure that the **Windows SQL** plug-in is selected in the **Restore Plug-in** list.
 - f. Select the encryption method to use for client/server data transfer during the restore.

The exact encryption technology and bit strength used for a client/server connection depends on a number of factors, including the client platform 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 134](#).
 - l. Disregard the remaining restore options, which do not apply when you are restoring 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 was initiated.
 - o. Click **Close**.
8. If necessary, restart the SQL Server Agent service:
 - To start the default instance of SQL Server Agent, open a command prompt and type:
net start SQLSERVERAGENT
 - To start a named instance of SQL Server Agent, open a command prompt and type:
net start SQLAGENT\$INSTANCE
where INSTANCE is the name of the instance.

Restoring a database in an AlwaysOn availability group

The steps to restore a database in an AlwaysOn availability group depend on whether the database is on the primary replica or a secondary replica.

Restoring a database on the primary replica

To restore a database on the primary replica:

1. Remove the database from the availability group.
2. Select the physical node that is hosting the primary replica as the Avamar client from which to restore. [“Backup requirements for AlwaysOn availability groups” on page 38](#) explains how to determine which node is hosting the primary replica.
3. Use the Avamar Plug-in for SQL Server to restore the database to its original location, as described in [“Restoring to the original location” on page 94](#).
4. Add the database back to the availability group.

NOTICE

After you remove the database from the availability group on the primary replica, the corresponding database on the secondary replicas are in a restoring state. There are two ways to restore the databases on the secondary replicas as part of the availability group. You can either delete the databases on the secondary replicas and then automatically re-create and synchronize the databases when you add the database back to the availability group, or you can 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. The SQL Server 2012 documentation available on the Microsoft website provides details.

Restoring a database on a secondary replica

To restore a database on a secondary replica:

1. Remove the database from the availability group.
2. Select the physical node that is hosting the primary replica as the Avamar client from which to restore. [“Backup requirements for AlwaysOn availability groups” on page 38](#) explains how to determine which node is hosting the primary replica.
3. Use the Avamar Plug-in for SQL Server to restore the database to the corresponding SQL Server instance on the other cluster node, as described in [“Restoring to a different instance” on page 101](#). During the restore, select NORECOVERY for the recovery operation in the plug-in options.
4. Join the database to the to the availability group as described in the SQL Server 2012 documentation on the Microsoft website.

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 that occurred since the most recent Avamar 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\FILENAME 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 using the instructions in [“How to find a backup by file or folder” on page 91](#).
3. In Avamar Administrator, restore the backup to its original location using the instructions in [“Restoring to the original location” on page 94](#).

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\FILENAME is the path and file name for the transaction log backup file.

5. If the database is not already listed 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.

The screenshot shows the 'Restore Command Line Options - Windows SQL' dialog box. It is divided into several sections:

- Select Plug-In Type:** A dropdown menu showing 'Windows SQL'.
- General Options:**
 - Use SQL REPLACE option
 - Tail-log backup
 - Recovery operation: **RECOVERY** (dropdown)
 - Standby file location:
 - Enable debugging messages
- System Databases:**
 - Restore system databases
 - Manage SQL services automatically during restore
- Redirected Restore:**
 - New database name:
 - Alternate database location:
 - Alternate log location: **Same as alternate database location** (dropdown)
 - Path to alternate log location:
- Authentication:**
 - SQL server address:
 - Authentication method: **NT authentication** (dropdown)
 - SQL login ID:
 - SQL password:
- Point-in-time Recovery:**
 - Point-in-time recovery mode: **None** (dropdown)
 - Point-in-time or mark name string:
 - Mark recovery point: **At mark** (dropdown)
 - Mark is after datetime:

At the bottom, there is a checked checkbox for 'Show Advanced Options', a 'More' button with a dropdown arrow, and 'OK', 'Cancel', and 'Help' buttons.

The following topics provide details on each of the restore options:

- ◆ “General restore options” on page 130
- ◆ “Recovery operation options” on page 131
- ◆ “System database restore options” on page 132
- ◆ “Redirected restore options” on page 133
- ◆ “Authentication options” on page 134
- ◆ “Point-in-time recovery options” on page 134

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. This specifies that SQL Server should 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 that is intended to prevent you from accidentally overwriting a different database or file. This safety check is described in the [Microsoft Transact-SQL Reference Manual](#) under 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. This may be required 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.
```

You should also use this option when you are restoring 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 process to capture the log records that have not been backed up. The tail-log backup is then used after the database restore to recover the transactions that were not included 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.

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 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 non-primary 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, and any transactions in the tail of the log may be lost.

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 for debugging purposes. Keep in mind that if you select the checkbox, very large log files are created. You should only use this option 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 7 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 necessary. The size of the file depends on the volume of undo actions resulting from uncommitted transactions. Specify the path to this file in the Standby file location box. The path should be in the following format:</p> <pre>DRIVE:\DIRECTORY\SUBDIRECTORY</pre> <p>where DRIVE is the drive letter on the destination client, and DIRECTORY\SUBDIRECTORY 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 specified location, then SQL Server overwrites it. If you do not specify a path, then Avamar creates the file in the C:\Program Files\avs\var directory, where C:\Program Files\avs is the Avamar installation directory.</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 all of 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 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 properly 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 are included in the restore. If you leave the checkbox clear, then the system databases are not restored.

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, and restarts the instance in single-user mode before the restore. After the restore, the instance is automatically restarted.
- ◆ 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 are restored first. You must select the **Manage SQL services automatically** during restore checkbox to ensure that all system databases are restored 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 are restoring 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. Choose the path to which to restore the 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 choose whether Avamar uses 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.

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 SQL server to connect to.
3. From the **Authentication method** list, choose 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 the Tail-log backup option is selected.

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, choose 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, choose 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.
5. 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.

Restore

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](#) 138
- ◆ [Performing disaster recovery](#) 138

Preparing for disaster recovery

To ensure that you are sufficiently prepared 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 using the Avamar Client for Windows. The backups should include:
 - File system
 - System state
 - Volumes

The *EMC Avamar for Windows Server User Guide* provides full backup instructions.

3. Back up all system and user databases using the Avamar Plug-in for SQL Server. [Chapter 3, “Backup,”](#) provides backup instructions.

NOTICE

If you don't perform the full Windows server backup and the database backups at the same time, then you must 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 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 121](#).
4. Use the SQL Server plug-in to restore the user databases as described in [“Restoring to the original location” on page 94](#).
5. When you are sure that the environment has been properly restored, 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](#) 140
- ◆ [Backup options.....](#) 140
- ◆ [Restore options.....](#) 143

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 for on-demand backup or restore operations, or when you create a dataset for a scheduled backup. You can set options by using the graphical controls and by typing options and values in the Enter Attribute and Enter Attribute Value fields.

NOTICE

No error checking or validation is performed on free text entries. In addition, free text entries override settings made using the graphical controls.

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 8 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:</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 has changed since 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 attempt to perform a differential or transaction log backup of the master database, a full backup is created instead.</p>
Enhanced data deduplication	<p>Enables or disables enhanced data deduplication. During backups, enhanced data deduplication typically reduces the amount of client data that must be sent to the server but requires additional client CPU resources. Choices are:</p> <ul style="list-style-type: none"> • Default — Use the global enhanced data deduplication setting already set on the server. This is the default setting. • Disabled — Do not use enhanced data deduplication for backups. • Enabled — Use enhanced data deduplication for backups.

Table 8 Backup plug-in options (page 2 of 4)

Setting	Description
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 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 necessary.</p> <p>Three backup sessions are created 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. Use the For simple recovery model databases list to control how Avamar handles transaction log backups of databases that use the simple recovery model.</p>
*Force full backup	<p>If selected, performs a full backup in the following scenarios:</p> <ul style="list-style-type: none"> • If you attempt to perform a differential or transaction log (incremental) backup and a full backup does not exist on the server. • If you attempt to perform a differential or transaction log backup after you restore a database. • If you attempt 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 in the event that no existing full backup is present on the Avamar server.</p>

Table 8 Backup plug-in options (page 3 of 4)

Setting	Description
*Truncate database log	<p>Controls database transaction log truncation behavior. One of the following:</p> <ul style="list-style-type: none"> • (Default) Only for incremental backup — The database transaction log is truncated if the backup type is set to incremental (transaction log). No log truncation occurs if the backup type is full or differential. • For all backup types — The database transaction log is truncated 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. • Never — The database transaction log is not truncated under any circumstances.
*For simple recovery model databases	<p>Specifies how Avamar handles 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	<p>Specifies to write maximum information to log files. If selected, very large log files are created.</p>
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 choose 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 6.</p> <p>If you specify only one stream, then Avamar backs up one database at a time with one stream.</p> <p>“Multi-streaming” on page 75 provides additional details.</p>
Minimum stream size	<p>Specifies the minimum size of each data stream.</p>
Authentication options	
*SQL server address	<p>Connect to this SQL server hostname or IP address.</p>

Table 8 Backup plug-in options (page 4 of 4)

Setting	Description
*Authentication method	Specifies whether to use NT authentication or SQL Server authentication to connect to SQL Server.
*SQL login ID	If SQL server authentication is selected in the Authentication method list, connect to the SQL server using this login ID.
*SQL password	If SQL server authentication is selected 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 9 Restore plug-in options (page 1 of 4)

Setting	Description
Use SQL REPLACE option	<p>If selected, an SQL WITH REPLACE clause statement is added to the restore Transact-SQL command. This 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 that is intended to prevent you from accidentally overwriting a different database or file. This safety check is described in the Microsoft Transact-SQL Reference Manual under the RESTORE command section.</p> <hr/> <p>This option might be required 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.</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 tail-log backup is then used after the database restore to recover the transactions that were not included in the backup.</p> <p>Tail-log backup is available only for databases that use the full and bulk-logged recovery models. It should be used for a direct restore without the SQL REPLACE option.</p> <p>Clear the checkbox if you are performing a redirected restore to a different SQL Server instance.</p>

Table 9 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 necessary. The size of the file depends on the volume of undo actions resulting from uncommitted transactions. Specify the path to this file in the Standby file location box. <p>STANDBY 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. 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>
*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. The path should be in the following format:</p> <p>DRIVE:\DIRECTORY\SUBDIRECTORY</p> <p>where DRIVE is the drive letter on the destination client, and DIRECTORY\SUBDIRECTORY 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 specified location, then SQL Server overwrites it. If you do not specify a path, then Avamar creates the file in the C:\Program Files\avs\var directory, where C:\Program Files\avs is the Avamar installation directory.</p>
*Enable debugging messages	<p>Specifies to write maximum information to log files. If selected, very large log files are created.</p>
System Databases options	
Restore system databases	<p>If selected, the plug-in creates an expanded list of databases to restore that includes system databases and non-system databases. By default, this option is cleared, and only non-system databases are expanded from the restore targets.</p>
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 is automatically restarted.</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 are restored in the proper order and with the necessary service stops and restarts.</p>
Redirected Restore options	

Table 9 Restore plug-in options (page 3 of 4)

Setting	Description
*New database name	To give the database that you are restoring a different name when you restore it to either a different instance on the same SQL server or to a different SQL 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) is restored 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 SQL server hostname or IP address.
*Authentication method	Specifies whether to use NT authentication or SQL Server authentication to connect to SQL Server.
*SQL login ID	If SQL server authentication is selected in the Authentication method list, connect to the SQL server using this login ID.
*SQL password	If SQL server authentication is selected 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 selected Point-in-time or Mark name from the Point-in-time recovery mode list, 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 9 Restore plug-in options (page 4 of 4)

Setting	Description
*Mark recovery point	<p>Specifies whether to include the mark in the recovery:</p> <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	<p>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.</p>

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](#) 148
- ◆ [Command reference](#) 151
- ◆ [CLI examples](#) 163

Understanding the SQL Server plug-in CLI

The following topics provide an overview of the SQL Server plug-in CLI.

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 directory. To launch the CLI, open a command prompt and change directory to the bin directory of the Avamar client installation directory. [“Command reference” on page 151](#) 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 is sorted 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 in the Backup Command Line Options dialog box when you perform an on-demand backup using Avamar Administrator, or on the Options tab when you create a dataset for a scheduled backup.

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 in the Restore Command Line Options dialog box when you perform a restore using 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` directory, where `C:\Program Files\avs` is the Avamar client installation directory. 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=FILEPATH` option, where `FILEPATH` 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 cluster

If SQL Server is installed in a clustering environment, the following steps are required:

- ◆ Run the CLI on the active node.
- ◆ When you use the CLI to browse, back up, or restore the local instance, the **--sqlserver** option is required, and you must specify the virtual server for the SQL Server.

Password encoding

You can use **avtar** to encode passwords that are entered through the command line or stored 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
```

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}

  [--backup-type=TYPE] [--exclude-pluginid-list=STRING |
  --excludepluginids=STRING] [--pluginid-list=STRING |
  --pluginids=STRING] [--version] [--sqltimeoutsecs=SECONDS]
  [--account=DOMAIN/CLIENT | --path=DOMAIN/CLIENT |
  --acct=DOMAIN/CLIENT] [--id=USER@DOMAIN/HOMEACNT]
  [--password=PASSWORD | --ap=PASSWORD | --pswd=PASSWORD]
  [--server=AVAMARSERVER | --hfsaddr=AVAMARSERVER]
  [--ntorsqlauth={ntauthentication | sqlauthentication}]
  [--pluginport=PORT] [--sqllogin=NAME] [--sqlpassword=PASSWORD]
  [--sqlserver=SQLSERVER] [--informationals=N] [--log]
  [--logfile=FILE] [--noinformationals] [--nostdout] [--nowarnings]
  [--quiet] [--verbose | -v] [--ddr={true | false}] [--ddr-index=N]
  [--encrypt=VALUE] [--encrypt-strength=VALUE] [--expires={DAYS |
  TIMESTAMP}] [--label=NAME] [--retention-type=TYPE |
  --retentiontype=TYPE] [--after-date=DATE] [--brtype={full |
  differential | incremental}] [--exclude=DATA] [--forcefull]
  [--forceincremental={true | false}] [--include=DATA]
  [--max-parallel=N] [--min-snapup-size=MB] [--prefix=VALUE]
  [--skip_or_prom_sdm={skip_with_error | skip_with_warning |
  promote_to_full}] [--truncatelog_ts={default | enable | disable}]
  [--validate] [--labelnum=NUMBER] [--after-date=DATE]
  [--aftertime=DATETIME] [--altlog={samelocat | difflocat}]
  [--app-instance=INSTANCE] [--dblocation=PATH]
  [--handle-sql-services-automatically |
  --handlesqlservicesautomatically]
  [--history-optimization={true | false}] [--loglocation=PATH]
  [--logtail] [--metadata] [--newdbname=NAME] [--pitormark={none |
  pointintime | markname}] [--pitormarkstr={PIT | MARK}]
  [--prefix=VALUE] [--recoveryoperation={RECOVERY | NORECOVERY |
  STANDBY}] [--recoverypoint={beforemark | atmark}]
  [--redirect=STRING] [--redirecttofile={true | false}]
  [--restore-destination={original | multiple | single}]
  [--restoresystem] [--standbyfilelocation=PATH] --target={PATH |
  CLIENT\INSTANCE} [--use-sql-replace-option |
  --usesqlreplaceoption]
```

Operations

Supply one and only one of the following operations for **avsql** using the **--operation** option.

Table 10 Operations for **avsql**

Operation	Description
browse	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 avsql displays output for only the first target.
backup	Performs an on-demand backup of the specified data.
restore	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 11 Common **avsql** options

Option	Description
--backup-type=TYPE	Internal use only.
--exclude-pluginid-list=STRING --excludepluginids=STRING	Internal use only.
--pluginid-list=STRING --pluginids=STRING	Internal use only.
--version	Displays the build version of the Avamar Plug-in for SQL Server.
--sqltimeoutsecs=SECONDS	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 12 Account options for **avsql** (page 1 of 2)

Option	Description
--account=DOMAIN/CLIENT --path=DOMAIN/CLIENT --acnt=DOMAIN/CLIENT	Required for all backup and restore operations. Specifies the client to back up or restore from using the following format: DOMAIN/CLIENT where DOMAIN is the Avamar domain to which the client belongs and CLIENT is the name of the client computer.
--id=USER@DOMAIN/CLIENT	Required for all backup and restore operations. Specifies the Avamar username for authentication. Note: You can specify the values for the --id and --account options at the same time by specifying --id=USER@DOMAIN/CLIENT.
--password=PASSWORD --ap=PASSWORD --pswd=PASSWORD	Required for all backup and restore operations. Specifies the password for the Avamar account.
--server=AVAMARSERVER --hfsaddr=AVAMARSERVER	Required for all backup and restore operations. Specifies the hostname or IP address of the Avamar server.
--ntorsqlauth={ntauthentication sqlauthentication}	Specifies whether to use NT authentication or SQL Server authentication to connect to SQL Server. One of the following values: <ul style="list-style-type: none"> • ntauthentication • sqlauthentication If not specified, then the default value of ntauthentication is used.

Table 12 Account options for **avsql** (page 2 of 2)

Option	Description
--pluginport=PORT	Specifies the port number on which to connect to the agent. If not specified, then the default value of 28002 is used.
--sqllogin=NAME	If you set --ntorsqlauth=sqlauthentication, then this option specifies the login ID for the SQL login account.
--sqlpassword=PASSWORD	If you set --ntorsqlauth=sqlauthentication, then this option specifies the password for the SQL login account.
--sqlserver=SQLSERVER	Hostname or IP address of the SQL server to connect to. If the SQL server is in a cluster, specify the virtual server name. Notice: This option is required when you use the CLI to browse, back up, or restore data in the local instance when the SQL server is in a cluster.

Logging options

The following logging options are available for **avsql**.

Table 13 Logging options for **avsql**

Option	Description
--informationals=N	Sets the information level for status messages, where N is a number such as 0, 1, 2, and so on.
--log	Redirects and appends output to an alternative log file specified by the --logfile=FILE option.
--logfile=FILE	Used with the --log option to specify the full path and file name of the alternative log file.
--noinformationals	Disables all status messages.
--nostdout	Disables output to STDOUT. However, if --log and --logfile=FILE are supplied, then output still goes to the log file.
--nowarnings	Disables warning messages.
--quiet	Suppresses all debugging messages.
--verbose -v	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 14 Backup options for **avsql** (page 1 of 4)

Option	Description
--ddr={true false}	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 true.
--ddr-index=N	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.
--encrypt=VALUE	Specifies the encryption method for client/server data transfer during the backup. One of the following values: <ul style="list-style-type: none"> • proprietary • tcp • ssl • sslverify • tls • tls-sa Use this option with the --encrypt-strength option. The <i>EMC Avamar Product Security Guide</i> provides details on how to properly specify the --encrypt and --encrypt-strength option values together to ensure proper encryption settings.
--encrypt-strength=VALUE	Specifies the encryption strength for client/server data transfer during the backup. One of the following values: <ul style="list-style-type: none"> • cleartext • medium • high Use this option with the --encrypt option. The <i>EMC Avamar Product Security Guide</i> provides details on how to properly specify the --encrypt and --encrypt-strength option values together to ensure proper encryption settings.
--expires={DAYS TIMESTAMP}	Specifies backup expiration as a number of days from today (DAYS) or an absolute TIMESTAMP.
--label=NAME	Assigns this descriptive label to the backup.
--retention-type=TYPE, --retentiontype=TYPE	Assigns advanced retention to the backup. One of the following values: <ul style="list-style-type: none"> • none — Do not explicitly assign any retention type to this backup. That is, treat the backup as a normal on-demand backup. • daily — Explicitly designate this backup as a daily backup. • weekly — Explicitly designate this backup as a weekly backup. • monthly — Explicitly designate this backup as a monthly backup. • yearly — Explicitly designate this backup as a yearly backup.
--after-date=DATE	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 mm/dd/yyyy format.

Table 14 Backup options for `avsql` (page 2 of 4)

Option	Description
<code>--brtype={full differential incremental}</code>	<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> <hr/> <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 attempt to perform a differential or transaction log backup of the master database, a full backup is created instead.</p>
<code>--exclude=DATA</code>	<p>Specifies an instance or database to exclude from the backup. When specifying the local instance, use “(local)”. For example:</p> <ul style="list-style-type: none"> • <code>--exclude=(local)</code> • <code>--exclude=(local)/DB1</code>
<code>--forcefull</code>	<p>Performs a full backup in the following scenarios:</p> <ul style="list-style-type: none"> • If you attempt to perform a differential or transaction log (incremental) backup and a full backup does not exist on the server. • If you attempt to perform a differential or transaction log backup after you restore a database. • If you attempt 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> <hr/> <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 in the event that no existing full backup is present on the Avamar server.</p>

Table 14 Backup options for `avsql` (page 3 of 4)

Option	Description
<code>--forceincremental={true false}</code>	<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 necessary.</p> <p>Two backup sessions are created 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.</p> <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/>
<code>--include=DATA</code>	<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 “(local)”. For example:</p> <ul style="list-style-type: none"> • <code>--include=(local)</code> • <code>--include=(local)/DB1</code>
<code>--max-parallel=N</code>	<p>Specifies the maximum number of streams to use for backup and restore. The default is 1, and the maximum value is 6. “Multi-streaming” on page 75 provides details.</p>
<code>--min-snapup-size=MB</code>	<p>Specifies the minimum size of each data stream in MB. The default value is 200.</p>

Table 14 Backup options for **avsql** (page 4 of 4)

Option	Description
--prefix=VALUE	Specifies a prefix for the log file name.
--skip_or_prom_sdm={skip_with_error skip_with_warning promote_to_full}	<p>Specifies how Avamar handles 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_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_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_to_full — A full backup occurs automatically instead of a transaction log backup for databases that use the simple recovery model. <p>The default value is skip_with_error.</p>
--truncatelog_ts={default enable disable}	<p>Specifies database transaction log truncation behavior. One of the following:</p> <ul style="list-style-type: none"> • default — The database transaction log is truncated if the backup type is set to incremental (transaction log). No log truncation occurs if the backup type is full or differential. • enable — The database transaction log is truncated 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. • disable — The database transaction log is not truncated under any circumstances.

Restore options

The following **avsql** options are available for the restore operation.

Table 15 Restore options for **avsql** (page 1 of 6)

Option	Description
--label=NAME	Specifies the label of the backup to restore. Use either this option or the --labelnum option during the restore.
--labelnum=NUMBER	Specifies the label number of the backup to restore. Use either this option or the --label option during the restore.
--after-date=DATE	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 mm/dd/yyyy format.

Table 15 Restore options for **avsql** (page 2 of 6)

Option	Description
--aftertime=DATETIME	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 yyyy-mm-ddThh:mm:ss format. For example, 2013-10-25T14:15:45 is October 25, 2013 at 2:15:45 p.m.
--altlog={samelocat difflocat}	Specifies whether the database log file (*.ldf) 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"> • samelocat • difflocat The default value is samelocat.
--app-instance=INSTANCE	When restoring to a different instance, this option specifies the target instance for the restore.
--dblocation=PATH	If you are relocating the database, this option specifies the full path of the new database file (*.mdf) location on the client, where PATH is a value such as c:\temp.
--encrypt=VALUE	Specifies the encryption method for client/server data transfer during the restore. One of the following values: <ul style="list-style-type: none"> • proprietary • tcp • ssl • sslverify • tls • tls-sa Use this option with the --encrypt-strength option. The <i>EMC Avamar Product Security Guide</i> provides details on how to properly specify the --encrypt and --encrypt-strength values together to ensure specific encryption settings.
--encrypt-strength=VALUE	Specifies the encryption strength for client/server data transfer during the restore. One of the following values: <ul style="list-style-type: none"> • cleartext • medium • high Use this option with the --encrypt option. The <i>EMC Avamar Product Security Guide</i> provides details on how to properly specify the --encrypt and --encrypt-strength values together to ensure specific encryption settings.
--handle-sql-services-automatically --handlesqlservicesautomatically	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 are restored in the proper order and with the necessary service stops and restarts.
--history-optimization={true false}	Enables or disables history search optimization, which you can further refine by using the --after-date=DATE option. The default setting is --history-optimization=true, which enables history search optimization.

Table 15 Restore options for **avsql** (page 3 of 6)

Option	Description
--loglocation=PATH	If <code>altlog=difflocat</code> , then this option specifies the path of the new database log file location on the client, where PATH is a value such as <code>c:\temp\logs</code> .
--logtail	Backs up the tail of the transaction log to capture the log records that have not been backed up. The tail-log backup is then used after the database restore to recover the transactions that were not included in the backup. Tail-log backup is available only for databases that use the full and bulk-logged recovery models. It should be used for a direct restore without the SQL REPLACE option. Do not use this option if you are restoring to a different SQL Server instance.
--metadata	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. You must use the --labelnum option to specify the backup with this option. The command fails if you use the --label option to specify the backup.
--newdbname=NAME	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 SQL server or to a different SQL server.
--pitormark={none pointintime markname}	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"> • none — Do not recover to a specific point in time. • pointintime — Recover to a specific date and time specified by --pitormarkstr=VALUE. • markname — Recover to the named mark specified by --pitormarkstr=VALUE. The default value is none.
--pitormarkstr={PIT MARK}	If --pitormark=pointintime or markname, then this option specifies the PIT (date and time) or named MARK to recover to. To recover to a date and time, use yyyy-mm-ddThh:mm:ss format. For example, 2013-10-25T14:15:45 is October 25, 2013 at 2:15:45 p.m.
--prefix=VALUE	Prefix for the name of the logfiles.

Table 15 Restore options for **avsql** (page 4 of 6)

Option	Description
<code>--recoveryoperation={RECOVERY NORECOVERY STANDBY}</code>	<p>Specifies the recovery operation that occurs after the restore. One of the following values:</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. <p>This option creates a file with recovery changes. You can use the file to revert the recovery changes, if necessary. The size of the file depends on the volume of undo actions resulting from uncommitted transactions. Specify the path to this file using the <code>--standbyfilelocation=PATH</code> option.</p> <p>STANDBY 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>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>
<code>--recoverypoint={beforemark atmark}</code>	<p>When performing 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"> • beforemark — 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. • atmark — 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>
<code>--redirect=STRING</code>	<p>Restores database files to the specified location. STRING contains the original file name and the destination path and file name:</p> <pre>--redirect=ORIGINAL_FILENAME=DESTINATION_PATH\NEW_FILENAME</pre> <p>To restore multiple files to multiple different locations, separate the STRING values with commas:</p> <pre>--redirect=ORIGINAL_FILENAME1=DESTINATION_PATH1\NEW_FILENAME1,ORIGINAL_FILENAME2=DESTINATION_PATH2\NEW_FILENAME2,ORIGINAL_FILENAME3=DESTINATION_PATH3\NEW_FILENAME3</pre> <p>To restore a file to its original location, specify only the original file name:</p> <pre>--redirect=ORIGINAL_FILENAME</pre>
<code>--redirecttofile={true false}</code>	<p>Specifies whether to restore database files as operating system files to the specified location.</p>

Table 15 Restore options for **avsql** (page 5 of 6)

Option	Description
--restore-destination={original multiple single}	<p>Specifies the restore destination for the data:</p> <ul style="list-style-type: none"> • original – Restore the data to its original location. This is the default value. • multiple – Restore the data to a different location. Use the ---target={PATH CLIENT\INSTANCE} option to specify the location. • single – Restore the data to operating system files. Use the ---target={PATH CLIENT\INSTANCE} option to specify the location.
--restoresystem	Attempts to restore databases that have the system attribute set.
--standbyfilelocation=PATH	<p>If --recoveryoperation=STANDBY, then this option specifies the path on the destination client to the standby file with recovery changes.</p> <p>The path should be in the following format:</p> <pre>DRIVE:\DIRECTORY\SUBDIRECTORY</pre> <p>where DRIVE is the drive letter on the destination client, and DIRECTORY\SUBDIRECTORY 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 specified location, then SQL Server overwrites it.</p> <p>You can use the file to revert the recovery changes, if necessary.</p>

Table 15 Restore options for **avsql** (page 6 of 6)

Option	Description
--target={PATH CLIENT\INSTANCE}	<p>Specifies the target location for the restored files:</p> <ul style="list-style-type: none"> • If --restore-destination=single, then specify the PATH on the original client to which to restore the files, such as --target=c:\sqldata. • If --restore-destination=multiple, then specify the INSTANCE on the destination client to which to restore the files. Use the following format: --target=CLIENT\INSTANCE where CLIENT is the name of the destination client and INSTANCE is the name of the instance on the client.
--use-sql-replace-option --usesqlreplaceoption	<p>Adds an SQL WITH REPLACE clause statement to the restore Transact-SQL command. This 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 that is intended to prevent you from accidentally overwriting a different database or file. This safety check is described in the Microsoft Transact-SQL Reference Manual under the RESTORE command section.</p> <p>This option might be required if a previous database restore exited with the following SQL Server error in the Avamar SQL restore log:</p> <p>One or more devices or files already exist. Reissue the statement using the WITH REPLACE option to overwrite these files and devices.</p> <p>You should also use this option when restoring system databases.</p>
--validate	Validates the integrity of a backup before performing the restore.

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 date the database was created, as well as the 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” on page 163](#)
- ◆ [“Backing up the local instance” on page 164](#)
- ◆ [“Backing up a named instance” on page 164](#)
- ◆ [“Backing up multiple instances” on page 164](#)
- ◆ [“Backing up a database” on page 165](#)
- ◆ [“Backing up multiple databases” on page 165](#)
- ◆ [“Backing up to a Data Domain system” on page 165](#)

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 " "  
--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
--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
"(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 instance named "sql02inst2" 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 --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 instance named "sql03inst2" 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"
"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
      "(local)/DB1" --label=DB1DIFF
```

The following command performs a full backup of the DB3 database in the instance named sql01inst2 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
      "SQLServer1\sql01inst2DB3/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=MCUser1 --brtype=full --path=/SQL/SQLServer1 --ddr=true
      --ddr-index=2 "SQLServer1\sql01inst2DB3/DB3" "(local)/DB1"
```

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 --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
      --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
      --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, use the `--path` option to specify the domain and client, and use the `--labelnum` option to specify the label number of the backup to restore.

The following command performs a tail-log backup and restores the backup with a label number of 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
```

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)
```

The following command performs a tail-log backup and restores the `DB3` database from the `Instance2` instance in the backup with a label number of 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 --labelnum=842
      Instance2/DB3
```

Restoring to a new database in the original instance

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 local instance from the backup labeled DB2Full 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
      --label=DB2Full (local)/DB2
```

Restoring to a different instance

To restore a backup to a different instance, 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`.

The following command restores the DB3 database from Instance2 in the backup with a label number of 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
```

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
```

You must run the previous command from a command line on the SQLServer2 (destination) client.

The following command restores the files with file names of File1 and File2 from the DB3 database in the local instance on the SQLServer1 client from their original location of C:\SQLData to the DB8 database in Instance2 on the SQLServer2 client with a new location of C:\SQL:

```
avsql --operation=restore --server=12.34.56.78 --id=AVAMARADMIN
      --ap=PASSWORD --path=/SQL/SQLServer1
      --target=SQLServer2\Instance2/DB8
      --redirect="C:\SQLData\File1"="C:\SQL\File1",
      "C:\SQLData\File2"="C:\SQL\File2" --labelnum=163 (local)/DB3
```

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 computer 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
```

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 computer 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
```

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, 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 with a label number of 395 to its original location of the SQLServer3 client in the SQL domain, including the system databases in the backup:

```
avsql --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
```

Point-in-time restore

To restore to a specific point in time, set `--pitormark=pointintime` and specify the date and time using 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 --pitormark=pointintime
      --pitormarkstr=2013-01-15T14:15:45 --label=FullBackup2 (local)/DB3
```

The following command restores the DB3 backup in Instance2 from the backup with a label number of 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 --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. You must use the `--labelnum` option to specify the backup with the `--metadata` option. The command fails if you use the `--label` option to specify the backup.

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
```

