EMC® Data Domain® Boost for Enterprise Applications
Microsoft Application Agent
Release 2.0

Installation and Administration Guide
P/N 302-001-438
REV 03
Contents

Performing the Microsoft application agent for SQL Server push backups by using the T-SQL script ................................................................. 57
Scheduling the Microsoft application agent for SQL Server backups .......... 58
  Using the SQL Server Agent to schedule the Microsoft application agent for SQL Server backups .............................................................. 58
  Using Windows Task Scheduler to schedule the Microsoft application agent for SQL Server backups ......................................................... 64
Performing the Microsoft application agent for SQL Server backups to a Data Domain Server over Fibre Channel ........................................ 64
Performing the Microsoft application agent for SQL Server backups to a Data Domain Server over backup LAN .............................................. 65

Chapter 5 Restore
Performing the Microsoft application agent for SQL Server restores .......... 68
  Using SSMS to perform the Microsoft application agent for SQL Server restores ..................................................................................... 68
  Using CLI to perform the Microsoft application agent for SQL Server restores ...................................................................................... 76
  Using the T-SQL script to perform the Microsoft application agent for SQL Server restores ................................................................. 79
Performing the Microsoft application agent for SQL Server push restores by using the T-SQL script ................................................................. 79
Performing the Microsoft application agent for SQL Server redirected restores ................................................................. 80
  Using SSMS to perform the Microsoft application agent for SQL Server redirected restores ................................................................. 80
  Using CLI to perform the Microsoft application agent for SQL Server redirected restores ................................................................. 81
Performing the Microsoft application agent for SQL Server disaster recovery 82

Appendix A Deleting the Microsoft Application Agent for SQL Server Expired Backups
Prerequisites........................................................................................................ 84
Using the expiry tool to delete the Microsoft application agent for SQL Server expired backups ................................................................. 85
  Command options and description.................................................................. 86
PREFACE

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

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

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

Purpose

This document describes how to install and configure Microsoft application agent for EMC Data Domain Boost for Enterprise Applications (DDBEA), and back up and restore the Microsoft applications.

Audience

This document is intended for the host system administrator, system programmer, or operator who is involved in installing and configuring Microsoft application agent for DDBEA and backing up and restoring the Microsoft applications.

Related documentation

The following EMC publications EMC Online Support (https://support.emc.com) provide additional information:

◆ EMC Data Domain Boost for Enterprise Applications Microsoft Application Agent Release 2.0 Release Notes
◆ EMC Data Domain Boost for Enterprise Applications and ProtectPoint Database Application Agent Release 2.0 Installation and Administration Guide
◆ EMC Data Domain Boost for Enterprise Applications and ProtectPoint Database Application Agent Release 2.0 Release Notes
◆ EMC Data Domain Operating System Administration Guide
◆ EMC Data Domain Operating System Initial Configuration Guide
◆ EMC Data Domain Operating System Command Quick Reference
◆ EMC Data Domain Operating System Release Notes
◆ EMC Data Domain Boost for Enterprise Applications Software Compatibility 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.

Typographical conventions

EMC uses the following type style conventions in this document:

**Bold**

Use for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks).

**Italic**

Use for full titles of publications referenced in text and for variables in body text.

**Monospace**

Use for:
- System output, such as an error message or script
- System code
- Pathnames, file names, prompts, and syntax
- Commands and options

**Monospace italic**

Use for variables.

**Monospace bold**

Use for user input.

[] Square brackets enclose optional values

| Vertical bar indicates alternate selections — the bar means “or”

{} Braces enclose content that the user must specify, such as x or y or z

... Ellipses indicate nonessential information omitted from the example

Where to get help

EMC support, product, and licensing information can be obtained as follows:

**Product information** — For documentation, release notes, software updates, or information about EMC products, go to EMC Online Support at:

https://support.emc.com

**Technical support** — Go to EMC Online Support and click Service Center. You will see several options for contacting EMC Technical Support. Note that 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.

**Online communities** — Visit EMC Community Network https://community.emc.com/ for peer contacts, conversations, and content on product support and solutions. Interactively engage online with customers, partners, and certified professionals for all EMC products.

Your comments

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

DPAD.Doc.Feedback@emc.com
Email your clarifications or suggestions for this document to:

DPAD.Doc.Feedback@emc.com

The following table presents the revision history of this document.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description of added or changed sections</th>
</tr>
</thead>
</table>
| 03       | May 13, 2016     | Updated the following sections:  
  • Prerequisites and installing the Microsoft application agent for SQL Server sections in the Installation and Configuration chapter  
  • Deleting the Microsoft Application Agent for SQL Server Expired Backups chapter  
  • Miscellaneous changes |
| 02       | December 16, 2015| Incorporated the technical review comments throughout the guide.                                        |
| 01       | September 28, 2015| First release of this document for EMC Data Domain Boost for Enterprise Applications Microsoft application agent release 2.0. |
CHAPTER 1
Overview

This chapter includes the following sections:

- Introduction to the Microsoft application agent .............................................. 10
- Features of the Microsoft application agent 2.0 for SQL Server ..................... 10
- Architecture of the Microsoft application agent for SQL Server ..................... 11
- Supported system requirements ........................................................................ 13
- Recommendations for better performance of backups and restores ............... 13
- Unsupported features and capabilities ............................................................... 13
Introduction to the Microsoft application agent

EMC® Data Domain® Boost for Enterprise Applications (DDBEA) Microsoft application agent is an application direct product that leverages an enterprise application data protection framework to provide an application centric data protection. The DD Boost modules integrate with the following technologies:

◆ The framework by using DD Boost
◆ Data Domain, a purpose-build backup appliance

The Microsoft application agent for DDBEA enables database and application administrators to efficiently back up and restore their applications by using the applications’ native tools. The users do not need much expertise to use the product.

The Microsoft application agent 2.0 supports only Microsoft SQL Server. The Microsoft application agent for SQL Server module integrates with the SQL Virtual Device Interface (VDI). The interface enables the user to configure the module by using a plug-in in the Microsoft SQL Server Management Studio (SSMS). The plug-in is similar to the SQL native backup and restore Graphical User Interface (GUI). Therefore, the database administrators (DBAs) can use the Microsoft native tools to backup and restore the SQL data.

NOTICE

Data Domain Boost devices do not distinguish between Transmission Control Protocol (TCP)/Internet Protocol (IP) and Fibre Channel (FC), and network types—LAN, WAN, and MAN. Data Domain Boost devices can successfully operate where packet loss is strictly 0% and latency is less than 20 ms.

NOTICE

The Microsoft application agent supports in-flight encryption for Data Domain Boost clients with a Data Domain 5.5 or later operating system over a WAN connection. To use this feature, you can configure the Data Domain system with either medium-strength or high-strength TLS encryption. The configuration is transparent to the Microsoft application agent. The latest EMC Data Domain Boost Administration Guide provides information.

Features of the Microsoft application agent 2.0 for SQL Server

The Microsoft application agent 2.0 for SQL Server provides the following new features and capabilities:

◆ Instance level backups
◆ Multiple database backups and restores
◆ Transact-SQL (T-SQL) scripting to perform backups and restores in a SQL environment
◆ Scheduled backups by using SQL Agent Jobs
◆ In case of the transaction log backups, either skip the simple model databases from the backups or promote the simple model databases to the full backups
◆ Copy full backups of the AlwaysOn Availability Group databases
◆ Support IPv6
Overview

- Copy or redirected pull restores from one SQL Server to another
- Scriptable return codes—0 and 1
- Backups and restores of databases with names that contain the following special characters in SQL standalone, cluster, and AlwaysOn Availability Group configurations:
  - ~
  - `'
  - !
  - @
  - %
  - ^
  - &
  - (  
  - )
  - -
  - _
  - {
  - }
  - \
  - ,

Architecture of the Microsoft application agent for SQL Server

The following figure illustrates the general architecture of the Microsoft application agent for SQL Server:

![Architecture Diagram](image-url)
Overview

The following figure illustrates the SQL Server - CLR Assembly architecture.

Figure 2  The Microsoft application agent for SQL Server - CLR Assembly architecture

The following are the important components of the Microsoft application agent for SQL Server:

- **SQL Server Management Studio plug-in**: The Microsoft application agent for SQL Server has a GUI which is a SSMS plug-in. The plug-in is similar to the SQL native backup and restore GUI.

- **Virtual Device Interface**: The Microsoft application agent for SQL Server utilizes VDI, which is an API provided by SQL Server, to integrate with the SQL Server and enables the Microsoft application agent to back up and restore SQL Server data. The Microsoft documentation provides information about the VDI technology. Run the Microsoft application agent on the same host that has the SQL Server.

- **DD Boost library**: Performs source-based deduplication and sends the backup data to the Data Domain Server.

- **EMC SQL-CLR Assembly**: CLR and T-SQL scripts must be integrated to create functions or procedures to perform backups and restores in a SQL environment. The Microsoft application agent for SQL Server installation deploys the EMC CLR assembly. The CLR assembly contains one exportable SQL function type routine to run any Microsoft application agent CLI command.
Supported system requirements

The *EMC Data Domain Boost for Enterprise Applications Software Compatibility Guide* provides information about system requirements such as, supported operating systems and applications.

Recommendations for better performance of backups and restores

Review the following recommendations for better performance of backups and restores of databases by using the Microsoft application agent for SQL Server:

- If the Data Domain system reports reaching a stream limit, reduce the number of stripes that the operations that are running around the time use.
- To ensure the consistency of the backups on the Data Domain system, configure all the backups of a SQL Server instance to use the same Data Domain system and path.
- Do not perform 5 or more parallel Microsoft application agent SQL-CLR jobs. For each SQL-CLR procedure or function, the SQL Server allocates a new thread from its thread pool and runs the SQL-CLR procedure or function in the context of that thread. The allocated thread is reclaimed in SQL server thread pool only after the SQL-CLR function or procedure is run. If you configure many SQL-CLR jobs to run at the same time, SQL Server allocates many threads from the same thread pool, and can eventually starve for available threads and other related resources. This affects the overall performance. Be cautious while deciding the number of parallel SQL-CLR jobs.

Unsupported features and capabilities

The Microsoft application agent for SQL Server does not support the following features and capabilities:

- A Data Domain user on the secondary Data Domain Server in a different group in which the primary Data Domain Server does not exist
- System database—master, model, and msdb restores
  - Use the `ddbsqlrc restore` command to restore the system databases.
  - “Using CLI to perform the Microsoft application agent for SQL Server restores” on page 76 provides information about the `ddbsqlrc` command.
- Tail-log backups of multiple databases during restore
- Copy restore of multiple databases of multiple instances
- Differential backups of the AlwaysOn Availability Group databases on secondary replicas
  - The differential backups are promoted to copy full backups.
- Tail-log backups on Data Domain Replicator (DDR) devices
- Instance level (MSSQL: save set) restores of databases by using T-SQL
- TDE-enabled databases for copy or directed restores
Overview

- Restoring the database files to a different path than the default path when you perform directed restores

By default, the SQL Server restores the database files to the SQL Server default datapath that you set in the SSMS.
CHAPTER 2
Installation and Configuration

This chapter includes the following sections:

◆ Prerequisites ............................................................................................................. 16
◆ Installing the Microsoft application agent for SQL Server ..................................... 17
◆ Uninstalling the Microsoft application agent for SQL Server ................................. 19
◆ Upgrading the Microsoft application agent for SQL Server ................................. 20
◆ Preparing the Data Domain System ......................................................................... 21
Prerequisites

Ensure that you meet the following prerequisites before you install the Microsoft application agent for SQL Server:

- You have installed the following applications on the host by running as an administrator:
  - Microsoft SQL Server
    “Supported system requirements” on page 13 provides information about the SQL versions that the Microsoft application agent supports.
  - Either management tools (SSMS) as part of the SQL Server installation or Microsoft Visual Studio 2010 Shell (Isolated) Redistributable Package.
  - .NET Framework 4.0

The EMC DD Boost for Enterprise Applications Software Compatibility Guide provides more information about the supported software and operating system versions.

- You have obtained one of the following software licenses:
  - Data Protection Suite for Enterprise Applications Capacity license
  - Data Protection Suite Capacity license

Contact the EMC sales representative for more details about the license for the environment.

**Note:** You receive a password with the software license. You must use the password when you uncompress the downloaded software package.

- You have downloaded the Microsoft application agent for DDBEA software package from EMC Online Support (https://support.emc.com) to the Windows Microsoft application agent for SQL Server host.

- For a SQL standalone server:
  - You have added the SQL Server user to the Administrators group in the Computer Management window.
  - You have set the User Account Control to Never notify in the Control Panel > User Accounts > Change User Account Control settings window.

- On Windows Server 2008 R2, before you perform a SQL Server 2012 or later backup, ensure that you have disabled the User Account Control for administrators.

  The Change the behavior of the User Account Control message for administrators in Admin Approval Mode section on the following page provides information.


- You must be an administrator both on the SQL host domain admin if the SQL host is part of domain and SQL server is running in the context of a domain admin, and SQL Server and master database to deploy Common Language Runtime (CLR) assembly and run CLR-integrated T-SQL functions in a SQL environment.
You have made .NET Framework 4.0 available in the MSSQL environment to register the CLR assembly that you require for T-SQL integration.

For the Windows login account that the Microsoft application agent backup and restore processes use to connect to a SQL Server, grant the following roles:

- SQL Server system administrator (sysadmin) role to provide the T-SQL backup query.
- SQL Server sysadmin role to open a shared memory handle when starting the VDI.

As a Microsoft application agent user, ensure that you are a member of the following groups:

- Local backup operators group
- Local administrators group

You have granted the SQL Server user (SERVICE) the administrator privileges to perform the Microsoft application agent for SQL Server backups to a Data Domain server over FC.

You have granted the SQL Server user (SERVICE) the write access to the Microsoft application agent for SQL Server installation folder and the log subfolder to successfully create backup and restore logs.

---

**Installing the Microsoft application agent for SQL Server**

**NOTICE**

The Microsoft application agent can coexist with other backup products that you use to protect data that the Microsoft application agent does not protect.

Perform the following steps to install the Microsoft application agent for SQL Server:

1. Use WinZip to uncompress one of the following downloaded files on the Microsoft application agent host:
   - `ddbma_win_x86.zip`, if you use an x86 host
   - `ddbma_win_x64.zip`, if you use an x64 host
   When you are prompted for a password, type the password that you received with the software license.

2. Use WinZip again to uncompress the resulting zip file.

3. Double-click one of the following setup files to start the installation:
   - `EMCDDDBMA.msi`, if you use a 32-bit host
   - `EMCDDDBMA64.msi`, if you use a 64-bit host

4. On the *Welcome to Microsoft app agent for DD Boost for Enterprise Apps Setup Wizard* page, click *Next*.

5. On the *End-User License Agreement* page:
   a. Select *I accept the terms in the License Agreement*.
   b. Click *Next*.
6. On the **Destination Folder** page, perform one of the following tasks and click **Next**.
   - Use the default installation location that appears in the text box.
   - Click **Change** to specify a different installation location in a dialog box that appears.
     In the dialog box, click **OK**.

7. On the **Ready to install Microsoft app agent for DD Boost for Enterprise Apps** page, click **Install**.

8. Review the information in “Deploying EMC CLR Assembly” on page 18 and on the **Deploying EMC CLR Assembly** page, click **Next**.

9. On the **Completed the Microsoft app agent for DD Boost for Enterprise Apps Setup Wizard** page, click **Finish**.

**Note:** Install the Microsoft application agent for SQL Server on all the nodes of a cluster so that CLR assembly is deployed for all the SQL Server instances.

If you have installed a SQL Server instance after you installed the Microsoft application agent, and you want to run T-SQL scripts on the this new instance, perform the following steps:

1. Rerun the Microsoft application agent for SQL Server installer in the **Repair** mode.
2. Proceed with the default selections on the wizard pages.
3. On the **Deploying EMC CLR Assembly** page, ensure that the CLR assembly is deployed on the new instance.

**Deploying EMC CLR Assembly**

You must integrate CLR and T-SQL scripts to create functions or procedures to perform backups and restores in a SQL environment.  

To deploy the CLR assembly on a SQL Server instance, you must enable or register the CLR assembly on the SQL Server. The Microsoft application agent for SQL Server installation process uses the registered CLR assembly to deploy the CLR-integrated T-SQL functions. The deployment creates a CLR assembly (Dynamic Link Library [DLL]) which contains one exportable SQL function type routine to run any Microsoft application agent for SQL Server CLI command. The routine requires one valid Microsoft application agent for SQL Server CLI command option.

“Using CLI to perform the Microsoft application agent for SQL Server backups” on page 53 and “Using CLI to perform the Microsoft application agent for SQL Server restores” on page 76 provide information about the Microsoft application agent for SQL Server CLI commands and supported options.

“Using SSMS to perform the Microsoft application agent for SQL Server backups” on page 42 and “Using SSMS to perform the Microsoft application agent for SQL Server restores” on page 68 provide information about how to create the T-SQL backup and restore scripts respectively by using the Microsoft application agent for SQL Server GUI.
You must deploy CLR functions on all the SQL Server instances on which you want to run the Microsoft application agent for SQL Server T-SQL scripts. If multiple SQL Server instances are running with different user accounts, you must run the Microsoft application agent for SQL Server installer as an admin user with CLR assembly deployment privileges for all the SQL Server instances. You must have sysadmin and serveradmin roles on each SQL Server instance.

If a SQL Server instance is shut down and you install the Microsoft application agent for SQL Server, the installation process does not deploy Common Language Runtime (CLR) assembly to the SQL Server instance. To deploy CLR assembly to the shut down SQL Server instance later when the instance is up and running, install the Microsoft application agent for SQL Server by running the installer in the changed mode.

Performing silent installation

Run the following command to install Microsoft application agent without any intervention:

```
msiexec /i <installation_path> /quiet /qn /norestart /l*v <log_file_location>
```

Example:

```
msiexec /i C:\EMCDDBMA_win_x64.msi /quiet /qn /norestart /l*v C:\install.log
```

Uninstalling the Microsoft application agent for SQL Server

Use one of the following methods to uninstall the Microsoft application agent for SQL Server:

- “Using the setup file” on page 19
- “Using the Windows uninstall program” on page 20
- “Using the command line interface (silent uninstallation)” on page 20

Using the setup file

Perform the following steps to uninstall the Microsoft application agent for SQL Server:

1. Double-click one of the following setup files to start the uninstallation process:
   - EMCDDDBMA.msi if you use a 32-bit host
   - EMCDDDBMA64.msi if you use a 64-bit host
2. On the Welcome to Microsoft app agent for DD Boost for Enterprise Apps Setup Wizard page, click Next.
3. On the Change, repair, or remove installation page, click Remove.
4. On the Ready to remove Microsoft app agent for DD Boost for Enterprise Apps page, click Remove.
5. On the EMC CLR assembly deployment wizard page, click Next.
6. On the Completed the Microsoft app agent for DD Boost for Enterprise Apps Setup Wizard page, click Finish.
Using the Windows uninstall program

Perform the following steps to uninstall the Microsoft application agent for SQL Server:

1. On the Windows desktop, click Start > Control Panel > Programs > Uninstall a program.
2. In the Uninstall or change a program window, perform one of the following steps:
   - Select Microsoft app agent for DD Boost for Enterprise Apps, and click Uninstall.
   - Double-click Microsoft app agent for DD Boost for Enterprise Apps.
     A confirmation message to uninstall the Microsoft application agent for SQL Server appears.
3. Click Yes.
4. On the EMC CLR assembly deployment wizard page, click Next.

Using the command line interface (silent uninstallation)

Run the following command to uninstall Microsoft application agent without any intervention:

```
msiexec /x <installation_path> /quiet /qn /norestart /l*v <log_file_location>
```

Example:

```
msiexec /x C:\0_Builds\Sub12-Build74\ddbma_win_x64\win_x64\EMCDDBMA64.msi /quiet /qn /norestart /l*v C:\install.log
```

Upgrading the Microsoft application agent for SQL Server

Perform the following steps to upgrade the Microsoft application agent for SQL Server:

1. Ensure that you meet all the requirements that the “Supported system requirements” on page 13 and “Prerequisites” on page 16 sections list.
2. Uninstall the previous version of the Microsoft application agent for SQL Server.
   The “Uninstalling the Microsoft application agent for SQL Server” on page 19 provides information.
3. Install the current version of the Microsoft application agent for SQL Server.
   The “Installing the Microsoft application agent for SQL Server” on page 17 section provides information.
Preparing the Data Domain System

Preparing the Data Domain System mainly includes the following tasks:

◆ “Creating storage units” on page 21
◆ “Configuring replication” on page 21

“Data Domain System Configuration” on page 23 provides detailed information about features, upgrading, configuring, and validating the Data Domain System.

Creating storage units

Perform the following steps to create storage units on the Data Domain Server to perform the Microsoft application agent for SQL Server backups and recoveries:

1. Log in to datadomain Enterprise Manager.
2. Click Data Management > DD Boost > Storage Units.
3. Click Create.
4. In the Create Storage Unit dialog box, perform the following steps to create a storage unit:
   a. In the Name field, specify the name of the storage unit to create.
      Each storage unit name must be unique and can contain 50 characters maximum with only 0-9, a-z, A-Z, n dash (-), and underscore (_).
   b. Click OK.
5. Repeat step 3 and step 4 to create multiple storage units.

“Creating storage units” on page 31 provides information about how to create storage units by using the command line interface.

The EMC Data Domain Operating System Administration Guide provides detailed information about storage units.

Configuring replication

Replicate data to remote Data Domain systems by using Data Domain Replicator. Replicating data enables you to perform recoveries in case of disasters.

Ensure that the replication process does not change the names of the folders and the files that the Microsoft application agent for SQL Server created on the storage node.

To enable the backup replication and subsequent restore from a secondary Data Domain system, the user ID or primary group ID of the DD Boost users on the primary and secondary systems must be identical.

The EMC Knowledgebase Article number 182294, titled Configuration of DDBoot Users on Source and Destination DDRs for MTree Replication, provides more information. The article is available on EMC Online Support (https://support.emc.com).

The Configuring replication section in the EMC Data Domain Operating System Administration Guide provides information about creating, enabling, disabling, and deleting replication pairs.
Installation and Configuration
CHAPTER 3
Data Domain System Configuration

This chapter includes the following sections:

- Features of Data Domain Boost ................................................................. 24
- Installing and upgrading the Data Domain operating system .................. 27
- Configuring the Data Domain system ....................................................... 28
- Validating the Data Domain system ......................................................... 38
Features of Data Domain Boost

DD Boost provides the following features:

- “Distributed segment processing” on page 24
- “Advanced load balancing and link failover” on page 25
- “Encrypted managed file replication” on page 27

“Configuring the Data Domain system” on page 28 provides information on how to configure these features.

Distributed segment processing

Distributed segment processing uses the DD Boost library on the database server and the Data Domain software on Data Domain Replicator. The Microsoft application agent for SQL Server loads the DD Boost library during backup and restore operations.

Distributed segment processing allows the Microsoft application agent for SQL Server to perform parts of the deduplication process, which avoids sending duplicate data to the Data Domain system that you configured as a storage server.

The distributed segment processing feature provides the following benefits:

- Increases throughput because the DD Boost library sends only unique data instead of all the data to the Data Domain system. The throughput improvements depend on the redundant nature of the data that you back up, the overall workload on the database server, and the database server capability. In general, greater throughput is attained with higher redundancy, greater database server workload, and greater database server capability.
- Decreases the network bandwidth requirements by sending the unique data to the Data Domain system through the network.

Manage distributed segment processing by using the `ddboost` command options. Data Domain recommends that you use distributed segment processing if the network connection is 1 Gb Ethernet. “Configuring distributed segment processing” on page 32 provides information on how to configure the distributed segment processing.

Distributed segment processing supports the following modes of operation for sending backup data to a Data Domain system:

- Distributed segment processing enabled
- Distributed segment processing disabled

Set the operation mode on the Data Domain system. The Microsoft application agent for SQL Server negotiates with the Data Domain system for the current setting of the option and accordingly performs backups.
Distributed segment processing enabled mode

When you enable the distributed segment processing feature, the DD Boost library performs the following tasks:

1. Segments the data.
2. Computes IDs for the data segments.
3. Checks with the Data Domain system for duplicate segments.
4. Compresses unique segments that the Data Domain system does not contain.
5. Sends the compressed data to the Data Domain system, which writes the unique data to disk.

You must configure the local compression algorithm that the DD Boost library uses on the Data Domain system. The EMC Data Domain Operating System Administration Guide provides more information about local compression and its configuration.

Distributed segment processing disabled mode

When you disable the distributed segment processing feature, the DD Boost library sends the data directly to the Data Domain system through the network. The Data Domain system then segments, deduplicates, and compresses the data before writing the data to the disk.

Note: You cannot disable the distributed segment processing feature on an Extended Retention Data Domain system.

Advanced load balancing and link failover

Note: This topic applies only if you use an Ethernet connection, not Fibre Channel, for backup and restore operations with the Microsoft application agent for SQL Server.

The advanced load balancing and link failover feature enables the following capabilities:

◆ Combination of multiple Ethernet links into a group
◆ Registration of only one interface on the Data Domain system with the Microsoft application agent for SQL Server.

If you configure an interface group, the Microsoft application agent for SQL Server negotiates with the Data Domain system on the registered interface to send the data. When the Data Domain system receives the data, the data transfer is load balanced and distributed on all the interfaces in the group.

The load balancing feature provides greater physical throughput to the Data Domain system as compared to configuring the interfaces into a virtual interface using Ethernet level aggregation.

The Data Domain system load balances the connections from multiple database servers on all the interfaces in the group. Advanced load balancing and link failover works at the DD Boost software layer. The feature is seamless to the underlying network connectivity, and supports both physical and virtual interfaces.
The feature balances the load of the data transfer depending on the number of the connections that are outstanding on the interfaces. The feature balances the load of only connections for backup and restore jobs.

The file replication connection between Data Domain systems is not part of the load balancing. You must use only one IP address for the target Data Domain system.

You must exclude one interface from the interface group (ifgroup) and reserve that interface for the file replication path between the source and target Data Domain systems.

Every installation of the Microsoft application agent for SQL Server must be able to connect to every interface that is a member of the interface group on the Data Domain system.

You can use the advanced load balancing and link failover feature with other network layer aggregation and failover technologies. You can put the links that connect the database servers and the switch that connects to the Data Domain system in an aggregated failover mode. This configuration provides end-to-end network failover functionality. You can use any of the available aggregation technologies between the database server and the switch.

The advanced load balancing and link failover functionality also works with other network layer functionality on the Data Domain systems, including VLAN tagging and IP aliasing. This functionality provides additional flexibility in segregating traffic into multiple virtual networks that run through the same physical links on the Data Domain system.

The EMC Data Domain Operating System Administration Guide provides more information about how to configure VLAN tagging and IP aliasing on a Data Domain system.

The advanced load balancing and link failover feature provides the following benefits:

◆ Eliminates the need to register one storage server for each host that runs the Microsoft application agent for SQL Server, which potentially simplifies the installation management.

◆ Routes the subsequent incoming backup jobs to the available interfaces if one of the interfaces in the group goes down while the Data Domain system is still operational.

◆ Increases link utilization by balancing the load of the backup and restore jobs on multiple interfaces in the group.

◆ Performs a transparent fail over of all in-flight jobs to healthy operational links when an interface fails. The process does not interrupt the jobs.

“Configuring advanced load balancing and link failover” on page 33 provides information about how to configure advanced load balancing and link failover.

Configuration restrictions

The Advanced load balancing and link failover feature has the following restrictions:

◆ You can add interfaces only to groups by using IP addresses.

◆ You must use the interfaces that have the same link speed in a group.

◆ You need a switch to connect multiple database servers because a Data Domain system supports one interface group only.
Encrypted managed file replication

By default, the database servers set up file replication jobs between two Data Domain systems without encryption after authenticating them by using the preconfigured username and password of DD Boost. If you enable the encrypted file replication feature, then when the database servers set up a replication job, the session between the source and destination Data Domain systems uses Secure Sockets Layer (SSL) to encrypt all image data and metadata sent over the WAN.

Enabling this feature on the Data Domain system is transparent to the Microsoft application agent for SQL Server. When the Microsoft application agent for SQL Server requests that the Data Domain system perform a file replication job, the source and destination systems negotiate to perform encryption without involving the Microsoft application agent for SQL Server. Encrypted file replication uses the ADH-AES256-SHA cipher suite. You cannot configure a different suite in Data Domain operating system. Enabling this feature does not require restarting the file system on a Data Domain system.

Encrypted file replication requires you to install a replicator license on both the source and the destination Data Domain systems that run DD OS 5.0 or later, and applies to all file replication jobs on the system.

You can use encrypted file replication along with the encryption of data-at-rest feature available on Data Domain operating system with the optional encryption license. When you use encrypted file replication with the encryption of data-at-rest feature, the backup process uses SSL to encrypt the encrypted backup image data over a WAN.

“Enabling encrypted file replication” on page 38 provides information on how to enable encrypted file replication. The *EMC Data Domain Operating System Administration Guide* provides more information about encrypted file replication.

Installing and upgrading the Data Domain operating system

The *EMC Data Domain Operating System Installation Guide* provide information on how to install and upgrade the Data Domain (DD) operating system.

You require a license to operate many of the features on a Data Domain system.

**Note:** You require the EMC DD Boost license to use the Microsoft application agent for SQL Server software. You also require a replication license for both the source and destination Data Domain systems to use the replication feature.

Contact the EMC Data Domain representative for more information and to purchase licensed features.

The *EMC Data Domain Operating System Administration Guide* provides details about all the licensed features and how to display and enable Data Domain licenses.
Configuring the Data Domain system

Perform the following tasks to configure the Data Domain system:

◆ “Opening ports in a firewall to enable Data Domain backups” on page 28
◆ “Enabling Data Domain Boost on a Data Domain system” on page 28
◆ “Configuring the Data Domain Boost server” on page 30

The EMC Data Domain Operating System Command Reference Guide provides complete descriptions of the commands used in these sections.

Opening ports in a firewall to enable Data Domain backups

Note: This topic applies only if you use an Ethernet connection, but not Fibre Channel, to perform backups and restores by using the Microsoft application agent for SQL Server.

Ensure that the following ports are open on the firewall to enable the DD Boost backups and optimized duplication.

◆ TCP 2049 (NFS)
◆ TCP 2051 (Replication)
◆ TCP 111 (NFS portmapper)
◆ TCP xxx (select a port for NFS mountd, where the default MOUNTD port is 2052)

Enabling Data Domain Boost on a Data Domain system

Every Data Domain system that supports DD Boost must have a unique name. You can use the DNS name of the Data Domain system, which is always unique.

Enable DD Boost on a Data Domain system by using one of the following methods:

◆ The ddboost enable command
◆ EMC Data Domain System Manager on the Data Management > DD Boost page that the EMC Data Domain Operating System Administration Guide describes

The following procedure includes commands that you can run at the operating system command line to enable DD Boost. The EMC Data Domain Operating System Command Reference Guide provides details about the commands.

Perform the following steps to enable Data Domain Boost:

1. On the Data Domain system, log in as an administrative user.
2. Run the following command to verify whether you have enabled the file system, and the file system is running:

   # filesystem status

   If the file system is disabled, run the following command to enable the file system:

   # filesystem enable
3. Run the following command to verify whether you have enabled the DD Boost license:

```
# license show
```

If the DD Boost license is disabled, run the following command to add the DD Boost license by using the license key that Data Domain provided:

```
# license add license-key
```

4. Establish the DD Boost username and password for the Data Domain system.

You can configure only one user for DD Boost access on a Data Domain system at a time. The username and password are case-sensitive.

Run the following commands to establish the username and password:

```
# user add username password password
# ddboost set user-name username
```

5. Run the following command to enable Data Domain Boost:

```
# ddboost enable
```

Changing Data Domain Boost access rights

When you enable the Data Domain Boost service for the first time on a Data Domain system, all the database servers can access the service by default. Use the `ddboost access` command to override this default, and restrict the access to specific database servers.

For example, you can run the following commands to remove the default access permission for all servers and add new access permissions for two specific database servers, `dbserver1.datadomain.com` and `dbserver2.datadomain.com`. The EMC Data Domain Operating System Command Reference Guide provides details about the commands.

- `# ddboost disable`
- `# ddboost access del clients *`
- `# ddboost access add clients dbserver1.datadomain.com dbserver2.datadomain.com`
- `# ddboost enable`

These commands establish the access controls that enable DD Boost access only to the two database servers—`dbserver1.datadomain.com` and `dbserver2.datadomain.com`.

**Note:** You must add the database server host that contains the Microsoft application agent for SQL Server software to a host access group before you configure backups.

If the commands do not establish any access controls, run the `ddboost enable` command to configure the default access control that enables all hosts to access the DD Boost service. If the commands have established any access controls, running the `ddboost enable` command enables the access controls. The `ddboost enable` command does not modify the access control list.
Consider the following guidelines when you change the Data Domain Boost access rights:

◆ Ensure that no backup operations are running to the Data Domain system. Run the `ddboost disable` command to prevent the backup operations.

Note: When you disable DD Boost, you disable the data access to all database servers.

◆ Specify only a fully qualified domain name, IP address, or resolvable DNS name for the client.

◆ If you have changed or deleted a username, the change in access rights does not affect any current operations. For example, deleting the current clients from the Data Domain Boost access list by running the `ddboost access del clients` command does not stop a backup that is in progress. All operations that are in progress continue. The current operations do not fail because of the change in access rights.

◆ After you change the access rights, run the `ddboost enable` command to re-enable Data Domain Boost and permit operations that are relevant to the changed access rights.

You can run the `ddboost access show` command to verify which database servers have DD Boost access rights. If the command output is `*`, all database servers have the access rights. For example:

```
# ddboost access show
DD Boost access allowed from the following clients
*
```

```
# ddboost access show
DD Boost access allowed from the following clients:
aehdb2
aehdb2.datadomain.com
aehdb3
aehdb3.datadomain.com
aehdb4
aehdb4.datadomain.com
aehdb5
aehdb5.datadomain.com

Run the following command to verify the active client connections:

```
# ddboost show connections
```

Configuring the Data Domain Boost server

Perform the following tasks to configure the Data Domain Boost server:

◆ “Creating storage units” on page 31

◆ “Configuring distributed segment processing” on page 32

◆ “Configuring advanced load balancing and link failover” on page 33

◆ “Configuring the DD Boost-over-FC service” on page 34

◆ “Enabling encrypted file replication” on page 38
Creating storage units

Create one or more storage units on each Data Domain system that you use with the Microsoft application agent for SQL Server. Ensure that you use a unique storage unit name on a single Data Domain system. However, you can use the same storage unit name on more than one Data Domain system.

**Note:** Storage unit names are case-sensitive.

You must provide the storage unit name when you perform the backup and restore operations with the Microsoft application agent for SQL Server.

You can create a storage unit by using one of the following methods:

- The `ddboost storage-unit` command
- **EMC Data Domain System Manager** on the Data Management > DD Boost page that the *EMC Data Domain Operating System Administration Guide* describes

Run the following command on the Data Domain system for each storage unit to create:

1. Run the following command on the Data Domain system:
   
   ```
   # ddboost storage-unit create storage_unit_name
   ```

2. Repeat step 1 for each Boost-enabled Data Domain system.

3. Run the following command to list the status of the storage units:
   
   ```
   # ddboost storage-unit show
   ```

You must create at least one storage unit on each Data Domain system that you will use with the Microsoft application agent for SQL Server. You can share a storage unit on a Data Domain system with more than one client system.

Deleting storage units

You can run the following command to delete a specified storage unit and its contents, and any DD Boost associations:

```
# ddboost storage-unit delete storage_unit_name
```

The `ddboost destroy` command deletes all storage units from the Data Domain system and permanently removes all the data files contained in the storage units.

The *EMC Data Domain Operating System Command Reference Guide* provides details about the `ddboost` commands.

Configuring quotas for storage units (optional)

Provision the storage on a Data Domain system through optional quota limits for a storage unit. You can specify quota limits either when you create a storage unit or later by using separate commands. You can specify quota limits at the storage unit level or the MTree level. The *EMC Data Domain Operating System Command Reference Guide* provides details about the `quota` and `ddboost` commands.

- Run the following command to enable quota limits on the Data Domain system:
   
   ```
   # quota enable
   ```
Data Domain System Configuration

- Run the following command to verify the quota status:
  
  ```bash
  # quota status
  ```

- Run the following command to configure quota limits when you create a storage unit:
  
  ```bash
  # ddboost storage-unit create storage_unit_name [quota-soft-limit n {MiB|GiB|TiB|PiB}] [quota-hard-limit n {MiB|GiB|TiB|PiB}]
  ```

- Run the following command to configure quota limits after you create a storage unit:
  
  ```bash
  # quota set storage-units storage_unit_list {soft-limit n {MiB|GiB|TiB|PiB}} {hard-limit n {MiB|GiB|TiB|PiB}}
  ```
  
  For example:
  
  ```bash
  # quota set storage-units SU_AEHDB5 soft-limit 10 GiB hard-limit 20 GiB
  SU_AEHDB5: Quota soft limit: 10240 MiB, hard limit: 20480 MiB
  ```

  Alternately, you can set the quota limits at the MTree level. For example:
  
  ```bash
  # quota set mtrees /data/coll1/SU_AEHDB5 soft-limit 10 GiB hard-limit 20 GiB
  /data/coll1/SU_AEHDB5: Quota soft limit: 10240 MiB, hard limit: 20480 MiB
  ```

- Run the following command to verify the quota limits of a storage unit:
  
  ```bash
  # quota show storage-units storage_unit_list
  # quota show mtrees mtree_path
  ```

Configuring distributed segment processing

You must configure the distributed segment processing option on the Data Domain system. The option setting applies to all the database servers and all the software that uses DD Boost.

You can manage the distributed segment processing by using one of the following methods:

- The `ddboost` command
- **EMC Data Domain System Manager** on the [Data Management > DD Boost](#) page that the *EMC Data Domain Operating System Administration Guide* describes

Run the following command to configure the distributed segment processing option:

```bash
# ddboost option set distributed-segment-processing {enabled | disabled}
```

Enabling or disabling the distributed segment processing option does not require a restart of the Data Domain file system.

A host on which you have installed the Data Domain Operating System (DD OS) release 5.2 or later enables the distributed segment processing feature by default. If you upgrade a host from DD OS release 5.0.x or 5.1.x to DD OS release 5.2 or later, the distributed segment processing feature remains in its previous state—enabled or disabled.
Configuring advanced load balancing and link failover

The advanced load balancing feature balances the load of data transfer and distributes the load in private network when the Data Domain system receives data from the DD Boost client. The process provides greater throughput especially for customers who use multiple 1 GbE connections.

The following restrictions apply to the configuration of the advanced load balancing and link failover feature:

- You can add Interfaces only to a group by using an IP address.
- You must not use the advanced load balancing and link failover feature with mixed 1GbE interfaces and 10 GbE interfaces in a group. Use interfaces with the same link speed in a group.

You can manage advanced load balancing and link failover by using one of the following methods:

- The `ddboost ifgroup` command
- EMC Data Domain System Manager on the Data Management > DD Boost page that the EMC Data Domain Operating System Administration Guide describes

Perform the following steps to create an interface group on the Data Domain system by adding existing interfaces to the group and registering the Data Domain system with the Microsoft application agent for SQL Server. After setting up the interface group, you can add or delete interfaces from the group. You must create the interfaces by using the `net` command before you create the interface group.

1. Run the `ddboost ifgroup` command to add the interfaces into the default group. For example:

   ```
   # ddboost ifgroup add interface 192.168.1.1
   # ddboost ifgroup add interface 192.168.1.2
   # ddboost ifgroup add interface 192.168.1.3
   # ddboost ifgroup add interface 192.168.1.4
   ```

   You can create only one interface group and you cannot rename this group.

2. Select one interface on the Data Domain system to register with the Microsoft application agent for SQL Server. Create a failover aggregated interface and register that interface with the Microsoft application agent for SQL Server. The EMC Data Domain Operating System Administration Guide describes how to create a virtual interface for link aggregation.

   It is not mandatory to use an interface in the ifgroup to register with the Microsoft application agent for SQL Server. You can use an interface that is not part of the ifgroup to register with the Microsoft application agent for SQL Server. You must register the interface with a resolvable name by using either DNS or any other name resolution mechanism.

3. Run the following command to enable the interface group on the Data Domain system:

   ```
   # ddboost ifgroup enable
   ```
4. Run the following command to verify the configuration:

```bash
# ddboost ifgroup show config interfaces
```

5. Add or delete interfaces from the group.

## Configuring the DD Boost-over-FC service

**Note:** If you do not use Fibre Channel (FC) to perform backups and restores by using the Microsoft application agent for SQL Server, skip this topic.

DD OS release 5.3 and later supports the Fibre Channel mechanism of communication between the Data Domain Boost library and the Data Domain system.

Before DD OS release 5.3, the user used IP networking for communications between the DD Boost library and any Data Domain system. The Microsoft application agent for SQL Server specifies the Data Domain system by using the hostname or IP address. The Data Domain Boost library uses TCP/IP connections to transmit requests to and receive responses from the Data Domain system.

DD OS release 5.3 introduces the alternative transport mechanism through Fibre Channel. Certain installations prefer or require the use of Fibre Channel as the data transfer mechanism between the Data Domain Boost library and Data Domain system. The Data Domain Boost over Fibre Channel transport (DD Boost-over-FC) enables such installations to access the features that the DD Boost technology provides.

Although Fibre Channel is specified as a general-purpose data transport mechanism, you can use Fibre Channel solely as a transport for SCSI device access in practice. Fibre Channel hardware and drivers reside solely within the SCSI protocol stacks in host operating systems. The DD Boost-over-FC transport must use SCSI commands for all communication.

To request access to a Data Domain system, the Microsoft application agent for SQL Server specifies the Data Domain system by using the special string DFC-\texttt{dfc\_server\_name}. \texttt{dfc\_server\_name} is the DD Boost-over-FC server name that is configured for the Data Domain system. The DD Boost-over-FC transport logic within the DD Boost library examines the set of generic SCSI devices available on the database server and uses SCSI commands to identify a catalog of devices which are pathnames of the SCSI devices that the database server operating system discovers.

The DD Boost-over-FC transport logic issues SCSI commands to the identified generic SCSI devices, to transfer Data Domain Boost protocol requests and responses between the library and the Data Domain system.

Most Data Domain Boost features are independent of transport. One notable exception is the DD Boost-over-IP advanced load balancing and link failover feature and its associated ifgroups. This feature is specific to the IP transport. You can achieve load balancing and link-level high availability for the DD Boost-over-FC transport through different means.

The DD Boost-over-FC communication path applies only between the database server/Data Domain Boost library and the Data Domain system, and does not apply to communication between two Data Domain systems.
To enable the support of the DD Boost-over-FC service, you must install the supported Fibre Channel target HBAs on the host. The *EMC Data Domain Operating System Command Reference Guide* and *EMC Data Domain Operating System Administration Guide* provide information about using the `scsitarget` command for managing the SCSI target subsystem.

Perform the following steps to configure the DD Boost-over-FC service:

1. Run the following command to enable the DD Boost-over-FC service:
   ```bash
   # ddboost option set fc enabled
   ```

2. Optionally, run the following command to set the dfc-server-name:
   ```bash
   # ddboost fc dfc-server-name set server_name
   ```
   Alternatively, accept the default name, which is the base hostname of the Data Domain system. A valid dfc-server-name consists of one or more of the following characters:
   - lowercase letters (a–z)
   - uppercase letters (A–Z)
   - digits (0–9)
   - underscore (_)
   - dash (–)
   
   **Note:** The dot or period character (.) is not valid within a dfc-server-name. You cannot use the fully qualified domain name of a Data Domain system as the dfc-server-name.

3. Run the following command to create a DD Boost FC group:
   ```bash
   # ddboost fc group create group_name
   ```
   For example:
   ```bash
   # ddboost fc group create lab_group
   ```

4. Run the following command to configure the device set of the DD Boost FC group:
   ```bash
   # ddboost fc group modify group_name device-set count count endpoint { all | none | endpoint_list }
   ```
   For example:
   ```bash
   # ddboost fc group modify lab_group device-set count 8 endpoint all
   ```

5. Run the following command to add initiators to the DD Boost FC group:
   ```bash
   # ddboost fc group add group_name initiator initiator_spec
   ```
   For example:
   ```bash
   # ddboost fc group add lab_group initiator "initiator-15,initiator-16"
   ```

6. Verify that the DFC devices are visible on the client.

7. Ensure that the user who performs the backups and restores has the required permission to access the DFC devices.
Managing the DD Boost-over-FC path

The ifgroup-based advanced load balancing and link failover mechanism does not apply to the Fibre Channel transport. The feature provides a different path mechanism for the DD Boost-over-FC solution.

The Data Domain system advertises one or more Processor-type SCSI devices to the database server over one or more physical paths. The database server operating system discovers the devices and makes them available to applications through a generic SCSI mechanism (SCSI Generic driver on Linux, SCSI Pass-Through Interface on Windows).

Consider the following example:

- The database server has two initiator HBA ports—A and B
- Data Domain System has two FC target endpoints—C and D
- You have configured Fibre Channel Fabric zoning so that both initiator HBA ports can access both FC target endpoints
- You have configured Data Domain system with a DD Boost FC group that contains the following components:
  - Both FC target endpoints on the Data Domain system
  - Both initiator HBA ports
  - Four devices (0, 1, 2, and 3)

In this example, the media server operating system might discover up to 16 generic SCSI devices; one for each combination of initiator, target endpoint, and device number:

/dev/sg11: (A, C, 0)
/dev/sg12: (A, C, 1)
/dev/sg13: (A, C, 2)
/dev/sg14: (A, C, 3)
/dev/sg15: (A, D, 0)
/dev/sg16: (A, D, 1)
/dev/sg17: (A, D, 2)
/dev/sg18: (A, D, 3)
/dev/sg19: (B, C, 0)
/dev/sg20: (B, C, 1)
/dev/sg21: (B, C, 2)
/dev/sg22: (B, C, 3)
/dev/sg23: (B, D, 0)
/dev/sg24: (B, D, 1)
/dev/sg25: (B, D, 2)
/dev/sg26: (B, D, 3)

When the Microsoft application agent for SQL Server requests the Data Domain Boost library to establish a connection to the server, the DD Boost-over-FC transport logic within the DD Boost library uses SCSI requests to build a catalog of these 16 generic SCSI devices. The SCSI devices are paths to access the DD Boost-over-FC service on the Data Domain System. As part of establishing the connection to the server, the DD Boost-over-FC transport logic provides the catalog of paths to the server.
Selecting the initial path

The server maintains statistics about the DD Boost-over-FC traffic over the various target endpoints and known initiators. During the connection setup procedure, path management logic in the server considers these statistics and then selects the path that the server uses to establish the connection, based on the following criteria:

◆ Distribute the connections across different paths evenly for queue-depth constrained clients. “Queue-depth constraints” on page 38 provides more information.

◆ Select the least busy target endpoint

◆ Select the least busy initiator from among paths to the selected target endpoint

Dynamic rebalancing

The server periodically performs dynamic rebalancing when the statistics reveal the following situations:

◆ For queue-depth constrained clients that “Queue-depth constraints” on page 38 describes, connections are distributed unequally across available paths

◆ Workload across target endpoints is out of balance

◆ Workload across initiators is out of balance

When the server finds one of these situations, the server marks one or more connections for server-directed path migration. In a future data transfer operation, the server requests that the DD Boost library start using a different available path from the catalog for subsequent operations.

Client path failover

The server dynamic rebalancing logic directs the client to use a different path. However, the client can also decide to use a different available path if the client receives errors while using the connection’s current path.

For example, assume the path catalog for a connection consists of eight paths:

/dev/sg21: (A, C, 0)
/dev/sg22: (A, C, 1)
/dev/sg23: (A, D, 0)
/dev/sg24: (A, D, 1)
/dev/sg25: (B, C, 0)
/dev/sg26: (B, C, 1)
/dev/sg27: (B, D, 0)
/dev/sg28: (B, D, 1)

The server selects the (A, C, 0) path during an initial path selection. The DFC transport logic in the DD Boost library starts sending and receiving data for the connection by using SCSI commands to /dev/sg21.

Later, the link from the target endpoint C to the switch becomes unavailable due to a cable disconnect or some hardware failure. Any subsequent SCSI request that the DFC transport logic submits to /dev/sg21 fails with an error code which indicates that the process could not deliver the SCSI request to the device.
In this case, the DFC transport logic looks in the catalog of devices for a path with a different physical component and a different combination of initiator and target endpoint. The DFC transport logic retries the SCSI request on the selected path, and repeats the process until the DFC transport logic finds a path that can successfully complete the SCSI request.

**Queue-depth constraints**

The specific SCSI device which receives a request is irrelevant for the DD Boost-over-FC solution. All SCSI devices are identical destination objects for SCSI commands as required by the SCSI protocol. When processing a SCSI request, the server logic gives no consideration to the specific device on which the SCSI request arrived.

Certain operating systems on clients impose a restriction on the number of outstanding I/O requests that the operating systems can process simultaneously over a SCSI device. For example, the Windows SCSI Pass-Through Interface mechanism conducts only one SCSI request at a time through each of its generic SCSI devices. This impacts the performance of the DD Boost-over-FC solution when multiple connections (for example, backup jobs) try to use the same generic SCSI device.

The Data Domain system also imposes a limit on the number of outstanding I/O requests for each advertised SCSI device. You must advertise multiple SCSI devices on the Data Domain system to overcome performance issues in case of heavy workloads. The term **queue-depth** describes the system-imposed limit on the number of simultaneous SCSI requests on a single device. Client systems such as Windows, the queue depth of which is so low as to impact performance, are considered to be queue-depth constrained.

**Enabling encrypted file replication**

Run the following command to enable the encrypted file replication option:

```
# ddboost file-replication option set encryption enabled
```

Enabling encrypted file replication requires additional resources—CPU and memory on the Data Domain system, and does not require a restart of the Data Domain file system. The *EMC Data Domain Operating System Administration Guide* provides information on encrypted file replication.

**Validating the Data Domain system**

Run the following commands to validate the status of the Data Domain system:

- `filesys status`
- `ddboost status`
- `ddboost ifgroup show config interfaces`
- `ddboost show connections`
- `ddboost storage-unit show compression`
- `ddboost storage-unit show`

The *EMC Data Domain Operating System Command Reference Guide* provides details about these commands and their options.
Run one of the following commands. The command that you use to validate the communication between the database server and the Data Domain system depends on the type of the network connection that you use.

- If you have a DD Boost-over-IP system, then log in to the database server, and run the following command:

  ```
  # rpcinfo -p Data_Domain_system_hostname
  ```

  The command output must include the ports listed in “Opening ports in a firewall to enable Data Domain backups” on page 28.

- If you have a DD Boost-over-FC system, then log in to the database server, and run the relevant command to verify whether the DFC devices are visible on the client.

  The *EMC Data Domain Operating System Command Reference Guide* provides details about the supported commands.

Use the `ddbadmin -G` command to verify the username, password, and valid access permissions for the Data Domain system.
Data Domain System Configuration
CHAPTER 4
Backup

This chapter includes the following sections:

◆ Performing the Microsoft application agent for SQL Server backups .................. 42
◆ Performing the Microsoft application agent for SQL Server push backups by using the T-SQL script .................................................................................................................. 57
◆ Scheduling the Microsoft application agent for SQL Server backups ..................... 58
◆ Performing the Microsoft application agent for SQL Server backups to a Data Domain Server over Fibre Channel ................................................................................ 64
◆ Performing the Microsoft application agent for SQL Server backups to a Data Domain Server over backup LAN .................................................................................... 65
Performing the Microsoft application agent for SQL Server backups

You can perform the Microsoft application agent for SQL Server backups in both the SQL standalone and cluster environments.

Use one of the following methods to back up either all the databases or only the required databases of a SQL Server instance:

- “Using SSMS to perform the Microsoft application agent for SQL Server backups” on page 42
- “Using CLI to perform the Microsoft application agent for SQL Server backups” on page 53
- “Using the T-SQL script to perform the Microsoft application agent for SQL Server backups” on page 56

**Note:** The procedures in this section apply to both the standalone and cluster environments of SQL.

Using SSMS to perform the Microsoft application agent for SQL Server backups

**NOTICE**

If you have not installed SQL Server Management Studio (SSMS), double-click the DDSSMSAddinApp.exe file in the \Microsoft_application_agent_for_SQL_Server_installed_folder\EMC DD Boost Modules\DDBMA\bin folder to start the Microsoft application agent for SQL Server. The Microsoft app agent for DD Boost for Enterprise Apps window appears as shown in Figure 4 on page 44.

Perform the following steps to back up a database by using SSMS:

1. On the Windows desktop, click **Start** > **Microsoft SQL Server Management Studio**.
The Microsoft SQL Server Management Studio window appears as shown in Figure 3 on page 43.
2. Click Microsoft app agent for DD Boost for Enterprise Apps.
   
   The Microsoft app agent for DD Boost for Enterprise Apps window appears as shown in Figure 4 on page 44. The Backup tab is selected by default.

   ![Microsoft app agent for DD Boost for Enterprise Apps window](image)

   **Figure 4** Microsoft app agent for DD Boost for Enterprise Apps window

3. On the **General** page, perform the following steps:
   
   a. Under **Source**, specify the following fields:
      
      - **SQL Server Instance**: Select the name of the SQL Server instance—standalone or cluster, where the databases to back up exist.
      
      - From the list, you can select either the entire SQL Server instance, that is, all the databases by selecting the check box in the header row or only the required databases to back up.

      - **Backup type**: Select the type of the backup—Full, Transaction log, or Differential, to perform.
**Note:** Do not select the SQL Server host option. To use this option, contact the EMC technical support professional.

b. Under **Backup set**, specify the following fields:
   - **Name**: Specify a name for the backup that you want to perform.
   - **Description**: Briefly describe the backup that you want to perform.
   - **Expires after (days)**: Specify the number of days after which the backup must expire.

c. Under **Destination**, click the button beside the **Data Domain Server** field to select the Data Domain Server to which you want to back up the database.

The **Data Domain Connection List & Lockbox Settings** dialog box appears as shown in Figure 5 on page 45.

![Data Domain Connection List & Lockbox Settings](image)

**Figure 5** Data Domain collection list and lockbox settings

Perform the following steps:

- From the **DataDomain Connections** list, select the Data Domain Server to which you want to back up the database.

  If the **DataDomain Server** list does not contain the Data Domain Server to which you want to back up the database, click **Add**.
The **Add Data Domain Server details** dialog box appears as shown in Figure 6 on page 46.

![Add Data Domain Server details](image1)

**Figure 6 Add Data Domain Server details**

Specify the following fields and click **OK** to add the Data Domain Server to the **DataDomain Server** list:

- **DataDomain Server**: Specify the name of the Data Domain server to which you want to back up the database.

- **Communication Protocol**: Select the medium—Ethernet or Fibre channel, through which you want to back up the database to the Data Domain Server. Ethernet is selected by default.

- **FC Service Name**: Specify the name of the Data Domain Server, which appears as shown in Figure 7 on page 46, as the FC service name if you have selected Fibre Channel from the **Communication Protocol** list.

![Data Domain Server name](image2)

**Figure 7 Data Domain Server name**

- **User Name**: Specify the username of the DD Boost user.

- **Password**: Specify the password of the DD Boost user.

- **Storage Unit**: Specify the name of the storage unit, to which you want to back up the selected databases.
Note: Ensure that you have specified valid values in these fields because there is no dynamic validation to ensure the correctness of the values.

If you want to remove a Data Domain server from the DataDomain Server list, select the server and click Remove.

– If you want to add SQL virtual servers to the lockbox, select Edit LockBox Settings.

Note: To either back up databases to SQL clustered instances or restore databases from SQL clustered instances by using the EMC T-SQL integration, you must add FQDNs of SQL virtual servers to the lockbox.

– To add a SQL virtual server to the lockbox, type the FQDN of the SQL virtual server in the Enter Host Name field and click Add.

– To remove a SQL virtual server from the lockbox, select the FQDN of the SQL virtual server from the Select Host Name list and click Remove.

– Click OK.

4. [Optional] Select Script and select one of the following options:

• CLI Scripts: To generate the CLI script. You can use the script to run a backup from the CLI.

• T-SQL Scripts: To generate a backup script in the T-SQL format. “Using the T-SQL script to perform the Microsoft application agent for SQL Server backups” on page 56 provides information about how to use the T-SQL scripts to perform the Microsoft application agent for SQL Server backups.

• [Optional] From the left pane, click Options to select the options that you require to perform the backup.

The page appears as shown in Figure 8 on page 48.
Select the options under the following fields according to your requirement:

- **Reliability:**
  - **Perform checksum before writing to media:** To perform a checksum operation with the backup and save the information to the backup media. The Microsoft application agent for SQL Server performs another checksum before a restore to ensure that the checksum matches the backup.
  
  - **Continue on error:** To use a checksum to detect a partial backup or restore state. The SQL Server verifies the checksum by calculating a local result and comparing the result with the stored value. If the values do not match and you encounter errors, you can select this option to continue the backup or restore operation.
• **Transaction log:**
  - **Truncate the transaction log:** To truncate the transaction logs before a backup. The Microsoft application agent for SQL Server enables this option if you have selected **Transaction log** as the backup type on the **General** page.
  - **Backup the tail of the log and leave database in restoring state:** To perform a tail-log backup of the database and leave the database in the restoring state.

• **Stripes:**
  
  **Create a striped backup:** To create a striped backup. Also, you can specify the number of stripes. The maximum limit is 32. If you specify a value beyond 32, the value defaults to 32.

• **Simple recovery model database:**
  - **Promote to full backup:** To perform the full backups of the SIMPLE recovery model databases because the SIMPLE recovery model databases do not support the transaction log backups. The Microsoft application agent for SQL Server enables this option if you have selected the SIMPLE recovery model databases to back up, and **Transaction log** as the backup type on the **General** page.
  - **Skip backup:** To skip backing up SIMPLE recovery model databases. The Microsoft application agent for SQL Server enables this option if you have selected the SIMPLE recovery model databases to back up, and **Transaction log** as the backup type on the **General** page. The SIMPLE recovery model databases do not support the transaction log backups.

• **AlwaysOn Group database replica:**
  - **Any replica:** To back up the selected databases that belong to any replica of the AlwaysOn Availability Group.
  - **Primary replica:** To back up the selected AlwaysOn Availability Group databases that are the primary replica and non-AlwaysOn Availability Group databases on the selected SQL Server instance.
  - **Secondary replica:** To back up the selected databases that belong to only the secondary replica of the AlwaysOn Availability Group and non-AlwaysOn Availability Group databases on the selected SQL Server instance.

In the **Availability Group Properties** window of SSMS, ensure that you have set the readability of the secondary replica to **Yes**, as shown in **Figure 9** on page **50:**
Figure 9  Availability Group Properties window

**Note:** SQL Server does not support full backups of the databases that belong to the secondary replica of the AlwaysOn Availability Group. If you have selected **FULL** as the backup type on the **General** page, the Microsoft application agent for SQL Server performs the **COPY_ONLY** backup instead of the full backup.

- **None (Don't Backup AAG Databases):** To skip backing up the selected databases that belong to any replica of the AlwaysOn Availability Group.

- **Others:**
  - **Skip databases that cannot be backed up at its current state:** To check the status of the selected databases and skip the databases that are not ready or available for the backup.

  If the status of the databases is **ONLINE**, the databases are ready or available for backups.

  If the status of the databases is **OFFLINE, EMERGENCY (SINGLE_USER Mode), SUSPECT, RESTORING, RECOVERING, or RECOVERY_PENDING**, the databases are not ready or available for backups.
- **Select a debug level**: To generate detailed logs that you can use to troubleshoot the backup issues. You can specify a value 1 through 9.

- **Data Domain over FC**: To back up the database to the Data Domain Server through Fibre Channel. You can back up to fibre channel only when you have properly configured the target Data Domain Server with FC, and correctly specified the information about the connection in the corresponding fields of the **Add Data Domain Server details** dialog box (Figure 6 on page 46).

  **Note**: Do not select the **Continue backup as set in 'backup type' and do not promote to full** option. To use this option, contact the EMC technical support professional.

5. **[Optional] Select Script** and select one of the following options:

   - **CLI Scripts**: To generate the CLI script. You can use the script to run a backup from the CLI.

   - **T-SQL Scripts**: To generate a backup script in the T-SQL format. “Using the T-SQL script to perform the Microsoft application agent for SQL Server backups” on page 56 provides information about how to use the T-SQL scripts to perform the Microsoft application agent for SQL Server backups.

   - Click **Run**.

The Monitor page displays the backup information and status as shown in **Figure 10** on page 52.
If the backup succeeds, the **Backup completed** message appears. Otherwise, the **Backup failed** message appears.

**Note:** Review the log files that are available in the `<Microsoft_application_agent_for_SQL_Server_installed_folder>\EMC DD Boost Modules\DDBMA\logs` folder for information about the success or failure of the backup.

**Note:** Restart the Microsoft application agent for SQL Server to ensure that the **Restore** tab displays the latest backup versions also, to perform recoveries.
Using CLI to perform the Microsoft application agent for SQL Server backups

The Microsoft application agent for SQL Server uses the following command to back up databases:

```
ddbmsqlsv [ <options> ] {path}
```

**Key options**

The following table lists the key backup command options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-N &lt;Backup_Set_Name&gt;</td>
<td>Specifies a name for the backup that you want to perform.</td>
</tr>
<tr>
<td>-b &lt;Backup_Set_Description&gt;</td>
<td>Describes the backup that you want to perform.</td>
</tr>
<tr>
<td>-S &lt;Number_of_Stripes&gt;</td>
<td>Creates a striped backup depending on the number of stripes that you specify. The maximum number of stripes that you can specify is 32. If you specify a value beyond 32, the value defaults to 32.</td>
</tr>
<tr>
<td>-l &lt;Backup_Type&gt;</td>
<td>Specifies the type of the backup—Full, Transaction log, or Differential, to perform.</td>
</tr>
<tr>
<td>-A &lt;Virtual_Server&gt;</td>
<td>Specifies the FQDN of the SQL virtual server to back up the databases to the SQL clustered instance by using the EMC T-SQL integration.</td>
</tr>
<tr>
<td>-c &lt;Client_or_Hostname&gt;</td>
<td>Specifies the SQL Server hostname that contains the SQL Server instance to back up.</td>
</tr>
<tr>
<td>-y &lt;Backup_Retention_Period&gt;</td>
<td>Specifies the number of days after which the backup must expire. The format is +&lt;Number_of_days&gt;d, where d indicates days. For example, -y +30d.</td>
</tr>
<tr>
<td>-a &quot;NSR_DFA_SI={TRUE</td>
<td>FALSE}&quot;</td>
</tr>
<tr>
<td>-a &quot;NSR_DFA_SI_USE_DD={TRUE</td>
<td>FALSE}&quot;</td>
</tr>
<tr>
<td>-a &quot;NSR_DFA_SI_DEVICE_PATH=&lt;Storage_Unit_Name&gt;&quot;</td>
<td>Specifies the name of the storage unit, to which you want to back up the selected databases.</td>
</tr>
<tr>
<td>-a &quot;NSR_DFA_SI_DD_HOST=&lt;Data_Domain_Server_Name&gt;&quot;</td>
<td>Specifies the name of the Data Domain server that contains the storage unit, to which you want to back up the databases.</td>
</tr>
<tr>
<td>-a &quot;NSR_DFA_SI_DD_USER=&lt;DDBoost_Username&gt;&quot;</td>
<td>Specifies the username of the DD Boost user.</td>
</tr>
<tr>
<td>-a &quot;NSR_DFA_SI_DD_PASSWORD=&lt;DDBoost_Password&gt;&quot;</td>
<td>Specifies the password of the DD Boost user.</td>
</tr>
</tbody>
</table>
Sample command to back up databases in a SQL standalone environment

dbmsqlsv.exe -c SQLX86.adesc.com -l full -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.196.90" -a "NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/sqlserver" "MSSQL$INST2005:d2"

Sample command to back up databases in a SQL cluster environment

dbmsqlsv.exe -c SQLcluster1.adesc.com -A SQLcluster1.adesc.com -l full -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.196.90" -a "NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/sqlserver" "MSSQL$INST2005:d2"

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| -a "NSR_INCLUDE_AAG_DATABASE={AnyReplica/Primary/Secondary/None}" | Specifies one of the following options of the AlwaysOn Availability Group databases to back up:  
  - Any replica: Backs up the selected databases that belong to any replica of the AlwaysOn Availability Group.  
  - Primary replica: Backs up the selected AlwaysOn Availability Group databases that are the primary replica on the selected SQL Server instance.  
  - Secondary replica: Backs up the selected databases that belong to only the secondary replica of the AlwaysOn Availability Group.  
  - SQL does not support full backups of the databases that belong to the secondary replica of the AlwaysOn Availability Group. If you have specified Full as the backup type, the Microsoft application agent for SQL Server performs the COPY_ONLY backup instead of the full backup.  
  - None (Don’t Backup AAG Databases): Skips backing up the selected databases that belong to any replica of the AlwaysOn Availability Group. |
| -a "SKIP_SIMPLE_DATABASE={TRUE | FALSE}" | Specifies whether to skip backing up SIMPLE recovery model databases. Use this option if you have selected the SIMPLE recovery model databases to backup, and specified Transaction log as the backup type. The SIMPLE recovery model databases do not support the transaction log backups. |
| -a "NSR_SKIP_NON_BACKUPABLE_STATE_DB={TRUE | FALSE}" | Specifies whether to check the status of the selected databases and skip the databases that are not ready or available for the backup.  
  - If the status of the databases is ONLINE, the databases are ready or available for backups.  
  - If the status of the databases is OFFLINE, EMERGENCY (SINGLE_USER Mode), SUSPECT, RESTORING, RECOVERING, or RECOVERY_PENDING, the databases are not ready or available for backups. |
Additional options

The following table lists the additional backup command options.

Table 2  Additional backup command options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-G</td>
<td>Specifies a NO_LOG transaction log backup before backing up the database. This option is valid only for full backups on SQL Server 2005.</td>
</tr>
<tr>
<td>-j</td>
<td>Performs a database consistency check before starting the backup.</td>
</tr>
<tr>
<td>-q</td>
<td>Displays ddbmsqlsv messages in the quiet mode, that is, the option displays summary information and error messages only.</td>
</tr>
<tr>
<td>-R</td>
<td>Uses the NO_TRUNCATE option when backing up transaction logs.</td>
</tr>
<tr>
<td>-T</td>
<td>Performs a TRUNCATE_ONLY transaction log backup before backing up the database. This option is valid only for full backups on SQL Server 2005.</td>
</tr>
<tr>
<td>-v</td>
<td>Displays ddbmsqlsv messages in the verbose mode, that is, the option provides detailed information about the progress of the backup operation.</td>
</tr>
<tr>
<td>-k</td>
<td>Performs checksum before backing up the data to the device.</td>
</tr>
<tr>
<td>-u</td>
<td>Performs checksum and continues the operation even in case of errors.</td>
</tr>
<tr>
<td>-l incr</td>
<td>Performs transaction log backup.</td>
</tr>
<tr>
<td>-H</td>
<td>Performs a tail-log backup of the database and leave it in the restoring state.</td>
</tr>
<tr>
<td>-D &lt;debug_level&gt;</td>
<td>Generates detailed logs that you can use to troubleshoot the backup issues.</td>
</tr>
<tr>
<td>-a &quot;NSR_ENABLE_FC=TRUE&quot;</td>
<td>Enables backing up the databases to the Data Domain Server through Fibre Channel.</td>
</tr>
<tr>
<td>-a &quot;NSR_FC_HOSTNAME=&lt;Data_Domain_Fibre_Channel_Server_Name&gt;&quot;</td>
<td>Specifies the Fibre Channel hostname.</td>
</tr>
</tbody>
</table>

Path

The path includes one of the following parameters:

- **Default instance or **d-path: MSSQL: or [MSSQL:]<s-path>[[MSSQL:]<s-path> [...]]

Where, s-path is the database

Example:

ddbmsqlsv -c NMMDA241.heroines.local -l full -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.77.27" -a "NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/heroines" "MSSQL:database1" "MSSQL:database2"
Backup

- **Named instance or i-path:** MSSQL$<Instance_Name> or MSSQL$<Instance_Name>:<s-path> [...] 

Where, *s-path* is the database

**Example:**

dbmsqlsv -c NMMDA241.heroines.local -l full -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.27.27" -a "NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DD_HOST=10.31.192.10" -a "NSR_INCLUDE_AAG_DATABASE=AnyReplica" -a "NSR_SKIP_NON_BACKUPABLE_STATE_DB=TRUE" "MSSQL$SqlInst1:database1" "MSSQL$SqlInst1:database2"

**Note:** Use the **Script** option in the **Backup** page in the Microsoft application agent for SQL Server GUI to generate the CLI equivalent script that you can use to perform scheduled backups, automation, and other tasks.

**Using the T-SQL script to perform the Microsoft application agent for SQL Server backups**

The Microsoft application agent for SQL Server enables you to generate SQL-CLR script that you can use to perform the Microsoft application agent for SQL Server backups in the context of a SQL Server similar to any T-SQL command or script.

The SQL-CLR backup command is **emc_run_backup**. You must provide a VARCHAR type parameters to this command. This command uses the same Microsoft application agent for SQL Server backup CLI options. So, you must have a detailed knowledge of the Microsoft application agent for SQL Server backup CLI options to flawlessly use the **emc_run_backup** command. “Using CLI to perform the Microsoft application agent for SQL Server backups” on page 53 provides information about the Microsoft application agent for SQL Server backup CLI options.

**Sample T-SQL backup script**

```sql
USE [master]
GO
DECLARE @returnCode int
EXEC   @returnCode = dbo.emc_run_backup ' -c WIN8SQLSP.sharepoint.com -S 8 -l full -h "Set1" -y +0d -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.192.10" -a "NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/ddsub7" -a "NSR_INCLUDE_AAG_DATABASE=AnyReplica" -a "NSR_SKIP_NON_BACKUPABLE_STATE_DB=TRUE" "MSSQL:":'
IF @returnCode <> 0
BEGIN
RAISERROR ("Fail!", 16, 1)
END
```

You can use any SQL Server standard interfaces such as, SSMS (Query window) and OSQL command prompt to run the SQL-CLR scripts.

You can use the SQL-CLR script to schedule the Microsoft application agent for SQL Server backups by using the SQL Server Agent. “Using the SQL Server Agent to schedule the Microsoft application agent for SQL Server backups” on page 58 provides information.

“Recommendations for better performance of backups and restores” on page 13 provides guidelines for better performance of the Microsoft application agent for SQL Server backups by using the SQL-CLR scripts.
Performing the Microsoft application agent for SQL Server push backups by using the T-SQL script

To back up either all the databases or only the required databases of a SQL Server instance from a source host to a Data Domain device by using a different host, perform the following steps:

1. Start SSMS on the different host that you use to perform the backup.
2. By using SSMS, connect to the source SQL Server instance on the source host to back up to the Data Domain device.
3. In the SSMS window on the different host, click New Query.
4. In the New Query window, run the T-SQL script to perform the backup.

You can either generate the T-SQL script by using the Microsoft application agent GUI on the source host and copy it to the New Query window on the different host or write the T-SQL script in the New Query window. “Using SSMS to perform the Microsoft application agent for SQL Server backups” on page 42 and “Using the T-SQL script to perform the Microsoft application agent for SQL Server backups” on page 56 provide information.

Sample T-SQL push backup script

```
USE [master]
GO
DECLARE @returnCode int
EXEC   @returnCode = dbo.emc_run_backup ' -c CLUST-SQL-02.contoso.com
-A CLUST-SQL-02.contoso.com -l full -y +0d -a "NSR_DFA_SI=TRUE" -a
"NSR_DFA_SI_DD_HOST=nmmddtwo.sp2010.com" -a
"NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/ddbmav2b75" -a
"NSR_INCLUDE_AAG_DATABASE=AnyReplica" -a
"NSR_SKIP_NON_BACKUPABLE_STATE_DB=TRUE"
"MSSQL$Inst1:Clus-SQL-01-DB01" "MSSQL$Inst1:CLUST-SQL-02-DB01"
"MSSQL$Inst1:CLUST-SQL-02-DB02" "MSSQL$Inst1:CLUST-SQL-02-DB03"
' IF @returnCode <> 0
BEGIN
RAISERROR ('Fail!', 16, 1)
END
```

Where:

- **CLUST-SQL-02.contoso.com** is the source host that has the source SQL Server instance to back up.
- **Inst1** is the source SQL Server instance, to which you must connect from the different host to perform the backup.
- **Clus-SQL-01-DB01, CLUST-SQL-02-DB01, CLUST-SQL-02-DB02, and CLUST-SQL-02-DB03** are the selected databases in the Inst1 instance to back up.
Scheduling the Microsoft application agent for SQL Server backups

Use one of the following methods to schedule the Microsoft application agent for SQL Server backups:

- “Using the SQL Server Agent to schedule the Microsoft application agent for SQL Server backups” on page 58
- “Using Windows Task Scheduler to schedule the Microsoft application agent for SQL Server backups” on page 64

Using the SQL Server Agent to schedule the Microsoft application agent for SQL Server backups

This section describes how to schedule the SQL Server Agent job for the Microsoft application agent for SQL Server backups for two subsystems—CmdExec and T-SQL. The SQL Server Agent is a job-scheduling agent which is contained in the SQL Server package. The SQL Server Agent's infrastructure consists of a Windows service that is used to execute jobs. The SQL Server Agent uses SQL Server to store job information. Each job contains one or more job steps. Each step contains its own task, for example, backing up a database. The SQL Server Agent can run a job on a schedule, in response to a specific event, or on demand.

Configuring the SQL Server Agent to schedule jobs

Configuring the SQL Server Agent comprises the following tasks:

- Enabling the SQL Server Agent
- Configuring security

Enabling the SQL Server Agent

The SQL Server Agent is in disabled state, by default. To enable the SQL Server Agent:

1. Open Object Explorer by using the View menu in SSMS.
2. Right-click SQL Server Agent and select Start.

Configuring security

To run the Microsoft application agent for SQL Server CLI commands in the CmdExec subsystem, you must have the administrator privileges. You can either change the SQL Server Agent service login credentials to the administrator or configure a proxy host to the CmdExec subsystem.

Changing the SQL Server Agent service login credentials

1. On your desktop, click Start \ Run....
2. In the Run dialog box, in the Open field, type services.msc and click OK.
3. In the Services window, right-click SQL Server Agent Service and select Properties.
4. In the **SQL Server Agent Service Properties** dialog box:
   a. On **Log On** tab, select **This account**.
   b. Type the credentials of the user, who has the administrator privileges, in the relevant fields.
      You can also click **Browse** and select the user who has the administrator privileges.
   c. Click **OK**.

**Creating proxy for the CmdExec subsystem**

The SQL Server Agent uses proxies to define the security context for job steps. A proxy is an object that enables the SQL Server Agent to access the stored credentials of a Windows user. When you run a job step that is configured to use a proxy, the SQL Server Agent uses the credentials that are defined in the proxy, and runs the job step by using the corresponding security context.

1. Open Object Explorer by using the **View** menu in SSMS.
2. Expand **Security**.
3. Right-click **Credentials** and select **Properties**.
4. On the **Credential Properties - EMC** page, specify the following fields and click **OK**:
   - **Credential name**: Type a name for the credential.
   - **Identity**: Type the name of the user who has the administrator privileges.
   - **Password**: Type the corresponding password for the user that you specified in the **Identity** field.
   - **Confirm password**: Retype the password that you specified in the **Password** field.
5. Create a proxy:
   a. Expand **SQL Server Agent**, and then expand **Proxies**.
   b. Right-click **Proxies** and select **New proxy**.
   c. On the **‘EMC’ Proxy Account Properties** page, specify the following fields and click **OK**:
      - **Proxy name**: Type a name for the proxy.
      - **Credential name**: Type the name of the credential that you specified in step 4.
      - **Active to the following subsystems**: Under this field, select **Operating system (CmdExec)**.

   **Note:** When you create a job step, you must select this proxy from the **Run as** list on the **Job Step Properties** page.
Scheduling a CmdExec job

You can schedule a CmdExec job on either a single SQL Server or multiple SQL Servers.

Scheduling a CmdExec job on a single SQL Server

Perform the following steps to schedule a CmdExec job:

1. Open Object Explorer by using the View menu in SSMS.
2. Expand SQL Server Agent.
3. Right-click jobs and select New job.
4. On the Job Properties window:
   a. On the General page, type in the Name, Owner, and Description fields.
   b. On the Steps page, click New... to create a step.
   c. On the Job Step Properties window:
      1. On the General page, specify the following fields:
         - Step name: Type a name for the job step.
         - Type: Select Operating system (CmdExec).
         - Run as: Depending on what you configured, select either proxy or SQL Server Agent Service Account.
         - Process execute exit code of a successful command: Type the process success exit code.
         - Command: Specify the required Microsoft application agent for SQL Server CLI command. “Using SSMS to perform the Microsoft application agent for SQL Server backups” on page 42 provides information about how to generate the command. The generated command does not have a full path to the binary. So, when you specify the generated command in this field, you must add the full path to the binary. If the binary path contains spaces, specify the path in the quotes.

         For example:
         
         "C:\Program Files\EMC DD Boost Modules\DDBMA\bin\ddbmsqlsv.exe" -D9 -c NMMDB154.nmmdev.com -l full -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.192.14" -a "NSR_DFA_SI_DD_USER=ddbma" -a "NSR_DFA_SI_DEVICE_PATH=/ddbma-sqlagent2" "MSSQL:master".

      2. On the Advanced page, specify the following fields:
         - On success action: Select an appropriate action—whether to proceed to the next job step or quit the current job step and report success, to be performed after the current job step succeeds.
         - Specify the other fields according to your requirements.
         - Click OK.
   d. On the Schedules page, click New... to schedule a job.
e. On the **Job Schedule Properties** window, specify the fields according to your requirements and click **OK**.

f. On the **Alerts** page, click **Add...** to create an alert to perform a job when a certain event occurs.

g. On the **New Alert** window:

1. On the **General** page, specify the following fields:
   - **Name**: Type a name for the alert.
   - **Type**: Select the type of the event.
   - Specify the other fields according to your requirements.

2. On the **Response** page, specify the following fields:
   - **Notify operators**: Select this option to notify, that is, send a message to, the operators about the job step status.
   - **New Operator**: Click this button to add an operator to the **Operator list**.

3. On the **Options** page, configure a method—E-mail, Pager, or Net Send to notify the operators about the status of the job step.

4. On the **Notifications** page, under **Actions to perform when the job completes**, select the appropriate notification methods to notify the status of the job step.

5. On the **Targets** page, select **Target local server**.

h. Click **OK**.

**Scheduling a CmdExec job on multiple SQL Servers**

Scheduling jobs from one central location to multiple target SQL Servers eases the database administrator’s job. To configure this set up, you must install the Microsoft application agent for SQL Server on all target machines, and make one SQL Server Agent a master and the rest targets.

**Configuring master and target SQL Server Agents**

1. Open Object Explorer by using the **View** menu in SSMS.

2. Right-click **SQL Server Agent** and select **Multi Server Administrator > Make this a Master...**

   The **Master Server Wizard** appears.

3. On the **Welcome to the Master Server Wizard** page, click **Next**.

4. On the **Master Server Operator** page, type in the required fields—E-mail address, Pager address, and Net send address, to notify the operators about the status of the job, and click **Next**.

5. On the **Target Servers** page:

   a. To specify the servers to use as targets for the SQL Server Agent jobs, under the **Registered servers** panel, select the servers and click the right arrow to move them to the **Target servers** panel.

   b. If you want to add servers to the **Registered servers** panel, click **Add Connection...**.
c. In the Checking Server Compatibility dialog box, click review the information and click Close.

The Master Server Login Credentials page appears.

6. Enable SQL Server remote connectivity:
   a. In SSMS, right-click the SQL Server and select Properties.
   b. In the Server Properties window:
      1. Under the Select a page panel, select Connections.
      2. Select Allow remote connections to this server.
      3. Click OK.

7. Specify general firewall exceptions on the SQL Server.

8. Set the encryption level on the target machines to 1 or 0 according to the security level if SSL certificate does not exist. To set the encryption level, change the registry entry of the following value to 1 or 0:

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\instance_name\SQLServerAgent\MsxEncryptChannelOptions


10. On the Complete the Wizard page, review the information and click Finish.

Running a job on the target servers

1. Perform step 1 through step 4g4 that “Scheduling a CmdExec job on a single SQL Server” on page 60 describes.

2. On the Targets page, select Target multiple servers, select the target servers from the list, and then click OK.

Scheduling a T-SQL job

T-SQL subsystem does not work under proxies.

Perform the following steps to schedule a T-SQL job:

1. Open Object Explorer by using the View menu in SSMS.

2. Expand SQL Server Agent.

3. Right-click Jobs and select New job.

4. On the Job Properties window:
   a. On the General page, specify the following fields:
      1. Name: Type a name for the job.
      2. Owner: Click the button beside the text box.
         In the Select Login dialog box, click Browse.
         In the Browse for Objects dialog box, under Matching objects, select NT SERVICE\SQLSERVERAGENT, and click OK.
         In the Select Login dialog box, click OK.
      3. Description: Type a description for the job.
b. On the **Steps** page, click **New...** to create a step.

c. On the **Job Step Properties** window:

1. On the **General** page, specify the following fields:
   - **Step name**: Type a name for the job step.
   - **Type**: Select **Transact-SQL script (T-SQL)**.
   - **Run as**: Depending on what you configured, select either proxy or **SQL Server Agent Service Account**.
   - **Process execute exit code of a successful command**: Type the process success exit code.
   - **Command**: Specify the required Microsoft application agent for SQL Server T-SQL command. "Using SSMS to perform the Microsoft application agent for SQL Server backups" on page 42 provides information about how to generate the T-SQL command.

You can run the generated T-SQL command by using the **New Query** menu option to check whether the operation succeeds. Successful run of the command ensures successful scheduled backups.

If you want to use return codes in the generated T-SQL command, you must modify the command.

**Sample raw T-SQL command**

```sql
USE [master]
GO
DECLARE @returnCode int
EXEC   @returnCode = dbo.emc_run_backup ' -c NNMDB154.nmmdev.com -l full -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.192.14" -a "NSR_DFA_SI_DD_USER=ddbma" -a "NSR_DFA_SI_DD_DEVICE_PATH=/ddbma-sqlagent2" -a "NSR_INCLUDE_AAG_DATABASE=AnyReplica" "MSSQL:Fabrics"' PRINT @returnCode
GO
```

**Sample T-SQL command with return codes**

```sql
DECLARE @returnCode int
EXEC   @returnCode = dbo.emc_run_backup ' -c NNMDB224.heroines.local -l full -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.77.27" -a "NSR_DFA_SI_DD_USER=ost112" -a "NSR_DFA_SI_DD_DEVICE_PATH=/heroines" -a "NSR_INCLUDE_AAG_DATABASE=AnyReplica" "MSSQL:CopyOftest_db_1" "MSSQL:db1"' IF @returnCode <>0 BEGIN RAISERROR ('Fail!', 16, 1) END
```

If the return code is error, the job step fails.

**Note**: The last two parameters—16 and 1 are necessary to raise an error if a job step fails.
Using Windows Task Scheduler to schedule the Microsoft application agent for SQL Server backups

Perform the following steps to schedule the Microsoft application agent for SQL Server backups:

1. On the Windows desktop, click Start > All Programs > Accessories > System Tools > Task Scheduler.
2. In the left pane in the Task Scheduler window, right-click Task Scheduler Library and select Create Basic Task.
   The Create Basic Task Wizard wizard appears.
3. On the Create a Basic Task page, specify a name for the task in the Name field and click Next.
4. On the Task Trigger page, select the appropriate option to start the task and click Next.
   The corresponding page of the selected option appears.
5. Specify or select all the fields that you require to perform the task and click Next.
6. On the Action page, select Start a program and click Next.
7. On the Start a program page, browse for the .bat file and specify its path in the Program/script field and click Next.
   Note: Step 3 to Step 5 in “Using SSMS to perform the Microsoft application agent for SQL Server backups” on page 42 provide information about how to create a .bat file.
8. On the Summary page, review the details of the task and click Finish.
   The process creates the task and adds it to your Windows schedule.

Performing the Microsoft application agent for SQL Server backups to a Data Domain Server over Fibre Channel

The procedure to back up a database to a Data Domain Server over FC is the same as the procedure described in “Using SSMS to perform the Microsoft application agent for SQL Server backups” on page 42. However, you must configure FC on the Data Domain Server. “Configuring the DD Boost-over-FC service” on page 34 provides information.

Note: The SQL Server user (SERVICE) must have the administrator privileges to perform the Microsoft application agent for SQL Server backups to a Data Domain server over FC.
Performing the Microsoft application agent for SQL Server backups to a Data Domain Server over backup LAN

If a Data Domain device has an interface on the backup LAN, the Microsoft application agent for SQL Server supports the backup LAN by default.

If the Data Domain device is connected only on the backup LAN, and the SQL host, on which the Microsoft application agent for SQL Server is installed, is multi-homed and has an interface on the backup LAN, the backups to the Data Domain device proceed over the backup LAN by default.

If both the Data Domain device and the SQL host are multi-homed, and are connected to the backup LAN, ensure that the Data Domain Server name that you specify in the Microsoft application agent for SQL Server GUI is the same as the backup LAN IP address. The backups to the Data Domain device proceed over the backup LAN.
Backup
CHAPTER 5
Restore

This chapter includes the following sections:

- Performing the Microsoft application agent for SQL Server restores ......................... 68
- Performing the Microsoft application agent for SQL Server redirected restores........ 80
- Performing the Microsoft application agent for SQL Server disaster recovery ........ 82
Performing the Microsoft application agent for SQL Server restores

You can perform the Microsoft application agent for SQL Server restores in both the SQL standalone and cluster environments.

Use one of the following methods to restore either an entire SQL Server instance or only the required databases:

◆ “Using SSMS to perform the Microsoft application agent for SQL Server restores” on page 68
◆ “Using CLI to perform the Microsoft application agent for SQL Server restores” on page 76
◆ “Using the T-SQL script to perform the Microsoft application agent for SQL Server restores” on page 79

Note: The procedures in this section apply to both the standalone and cluster environments of SQL.

Using SSMS to perform the Microsoft application agent for SQL Server restores

Perform the following steps to restore data by using SSMS:

1. On the Windows desktop, click Start > Microsoft SQL Server Management Studio.
2. Click Microsoft app agent for DD Boost for Enterprise Apps.
3. Click the Restore tab.
The page appears as shown in Figure 11 on page 69.

Figure 11  Restore page

4. On the General page, perform the following steps:
   
a. Under Sources, specify the following fields:
   
   - **DataDomain Server**: Click the button beside the field to select the Data Domain server and the storage unit to restore either SQL Server instance or required databases. A storage unit contains the backed up SQL Server hosts. In case of a remote (secondary) Data Domain server that has replicated databases to restore, select the relevant server.
     
     **Note**: A Data Domain user on the secondary Data Domain server must be in the same group as the primary Data Domain server.
   
   - **Data Domain over FC**: Select this option when you want to restore the database through Fibre Channel.
- **SQL Server host**: Select a SQL Server host to restore the required databases. A SQL Server host contains the backed up SQL Server instances—standalone or cluster and the corresponding databases.

  **Note**: To perform a redirected restore, select a SQL Server host that is not the current host.

- **SQL Server Instance**: Select the name of the SQL Server instance—standalone or cluster, to restore either entire instance or required databases. A SQL Server instance contains databases.

  The corresponding databases are displayed in the database list, which is below the **SQL Server Instance** field.

- From the database list, you can select either the entire SQL Server instance, that is, all the databases by selecting the check box in the header row or only the required databases to restore.

  The corresponding save sets are displayed in the save sets list, which is below the **Backup versions as on** field.

  The restore process restores the latest backup of each selected database. You can perform a Point-In-Time (PIT) restore if you have selected only one database.

  To restore a database that is part of the AlwaysOn Availability Group:
  
  a. Remove the database from the AlwaysOn Availability Group.
  
  b. In the databases list in the GUI, select the database that you have removed from the AlwaysOn Availability Group.
  
  c. Perform the restore as described in the subsequent steps and procedures.
  
  d. Add the restored database back to the AlwaysOn Availability Group.

- [Optional] You can select one of the backups that was performed within a specified timeline and perform restore.

  From the **Backup versions as on** list, select a date and time, and click **Show Versions**.

  All the backups that were performed within the specified timeline appear in the save sets list, which is below the **Backup versions as on** field.

  Select the save sets to restore.

  You can click << Older or Newer >> to browse for the save sets to restore.

  
  b. Under **Destination**, specify the following fields:

  - **SQL Server Instance**: Select the name of the destination SQL Server instance that must contain the selected SQL Server instance or databases to restore.
  
  - **Database**: Select the destination database to restore the selected SQL Server instance or databases.
Performing the Microsoft application agent for SQL Server restores

- **Restore to**: Click Timeline to specify the backup to restore. The **Backup Timeline** dialog box appears as shown in Figure 12 on page 71.

![Backup Timeline](image)

**Figure 12** Specifying the restore point

You can restore to either the last backup or any date and time.

Select the option and values, and click **OK**.

5. [Optional] From the left pane, click **Files** to change the default destination folders of the database files—data file (.mdf) and log file (.ldf).

The page appears as shown in Figure 13 on page 72.
The list on the page displays the locations of the database files when the database was backed up. If you do not want to restore the files to the same location, perform one of the following steps to change the destination paths:

- Select **Relocate all files to folder** and click the buttons on the right side of the **Data file folder** and **Log file folder** fields to specify the destination paths of the data files and the log file respectively. The operation restores all the data files to one folder, and the log file to another folder.

  **Note:** The Microsoft Application Agent for SQL Server does not support the relocated restore of multiple databases.

- Click the respective buttons on the extreme right side of the data files and the log file in the list to specify multiple folders to restore the files.
Performing the Microsoft application agent for SQL Server restores

6. [Optional] From the left pane, click **Options** to select the options that you require to perform the restore.

The page appears as shown in **Figure 14 on page 73**.

![Figure 14 Restore-Options page](image)

Select any of the following options:

- **Overwrite the existing database (WITH REPLACE)**: To overwrite the existing database with the current database that you restore, if the names of both the databases are same.

- **Recovery state**: To provide the following options and capabilities:
  - **RESTORE WITH RECOVERY**: To enable the ability to roll back uncommitted transactions and use the database to restore the latest or additional transaction logs.
- **RESTORE WITH NO RECOVERY**: To disable the ability to roll back uncommitted transactions and use the database to restore the latest or additional transaction logs.

- **RESTORE WITH STANDBY**: To enable the ability to undo committed transactions, saves the undo actions in a standby file that enables you to reverse the restore effects, and puts the database in the read-only mode. If you select this option, then specify the **Standby file** field by clicking the button on the right side of the field, browsing for the file, and selecting the file.

- **Select a debug level**: To generate detailed logs which you can use to troubleshoot the restore issues. Specify a value 1 through 9.

- **Take tail-log backup before restore**: To perform a transaction log backup of the data before performing restore. The tail-log backup ensures that the Microsoft application agent for SQL Server backs up the data that has changed from the previous backup.

  **Note**: The Microsoft Application Agent for SQL Server does not support the tail-log backup of multiple databases.

7. [Optional] Select **Script** and select one of the following options:

   - **CLI Scripts**: To generate the CLI script. You can use the script to run a backup from the CLI.

   - **T-SQL Scripts**: To generate a backup script in the T-SQL format. “Using the T-SQL script to perform the Microsoft application agent for SQL Server restores” on page 79 provides information about how to use the T-SQL scripts to perform the Microsoft application agent for SQL Server restores.

8. Click **Run**.

   The restore information and status appear in the **Monitor** page as shown in Figure 15 on page 75.
Performing the Microsoft application agent for SQL Server restores

If the restore succeeds, the **Restore completed** message appears. Otherwise, the **Restore failed** message appears.

**Note:** Review the log files that are available in the `<Microsoft_application_agent_for_SQL_Server_installed_folder>\EMC DD Boost Modules\DDBMA\logs` folder for information about the success or failure of the restore.
Using CLI to perform the Microsoft application agent for SQL Server restores

The Microsoft application agent for SQL Server uses the following command to restore databases:

```
dbmsqlrc [options] {path}
```

Key options

The following table lists the key restore command options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-t &lt;Last_Backup_Time_Strip&gt;</code></td>
<td>Specifies the last backup timestamp, to which you want to restore the selected databases.</td>
</tr>
<tr>
<td><code>-c &lt;Client_or_Hostname&gt;</code></td>
<td>Specifies the SQL Server hostname, to which you want to restore the required databases. A SQL Server host contains the backed up SQL Server instances—standalone or cluster and the corresponding databases. To perform a redirected restore, specify a SQL Server host that is not the current host.</td>
</tr>
<tr>
<td><code>-l &lt;Level&gt;</code></td>
<td>Generates detailed logs, which you can use to troubleshoot the restore issues. Specify a value 1 through 9.</td>
</tr>
<tr>
<td>`-d {&lt;Destination_Database_Name&gt;</td>
<td>MSSQL:&lt;Destination_Database_Name&gt;</td>
</tr>
<tr>
<td>`-s {MSSQL:&lt;database&gt;</td>
<td>MSSQL$&lt;Instance_Name&gt;:&lt;Database&gt;}`</td>
</tr>
<tr>
<td><code>-A &lt;Virtual_Server&gt;</code></td>
<td>Specifies the FQDN of the SQL virtual server to restore the databases from the SQL clustered instance by using the EMC T-SQL integration.</td>
</tr>
</tbody>
</table>
| `-S {normal | norecover | standby:<filepath>\undo.ldf}` | Performs one of the following tasks:  
  - normal: Enables the ability to roll back uncommitted transactions and use the database to restore the latest or additional transaction logs.  
  - norecover: Disables the ability to roll back uncommitted transactions and use the database to restore the latest or additional transaction logs.  
  - standby:<filepath>\undo.ldf: Enables the ability to undo committed transactions, saves the undo actions in a standby file that enables you to reverse the restore effects, and puts the database in the read-only mode. |
| `-a "NSR_DFA_SI={TRUE | FALSE}"` | Specifies whether the restore is NetWorker server-independent. The value must be TRUE. |
| `-a "NSR_DFA_SI_USE_DD={TRUE | FALSE}"` | Specifies whether the restore source is a Data Domain server. The value must be TRUE. |
| `-a "NSR_DFA_SI_DEVICE_PATH=<Storage_Unit_Name>"` | Specifies the name of the storage unit, from which you want to restore databases. |
Performing the Microsoft application agent for SQL Server restores

Sample command to restore databases in a SQL standalone environment

```
ddbmsqlrc.exe -c sqlx86.adesc.com -t "Monday, November 11, 2013 1:05:47 PM" -S normal -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.77.27" -a "NSR_DFA_SI_DD_USER=arti1" -a "NSR_DFA_SI_DEVICE_PATH=/artrep2" "MSSQL$SQL2K8:testddr1"
```

Sample command to restore databases in a SQL cluster environment

```
ddbmsqlrc.exe -c sqlcluster1.adesc.com -A sqlcluster1.adesc.com -t "Monday, November 11, 2013 1:05:47 PM" -S normal -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.77.27" -a "NSR_DFA_SI_DD_USER=arti1" -a "NSR_DFA_SI_DEVICE_PATH=/artrep2" "MSSQL$SQL2K8:testddr1"
```

Additional options

The following table lists the additional restore command options.

### Table 3  Key restore command options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a &quot;NSR_DFA_SI_DD_HOST=&lt;Data_Domain_Server_Name&gt;&quot;</td>
<td>Specifies the name of the Data Domain server that contains the storage unit, from which you want to restore databases. In case of a remote (secondary) Data Domain server that has replicated databases to restore, select the relevant server. A Data Domain user on the secondary Data Domain server must be in the same group as the primary Data Domain server.</td>
</tr>
<tr>
<td>-a &quot;NSR_DFA_SI_DD_USER=&lt;DDBoost_Username&gt;&quot;</td>
<td>Specifies the username of the DD Boost user.</td>
</tr>
<tr>
<td>-a &quot;NSR_DFA_SI_DD_PASSWORD=&lt;DDBoost_Password&gt;&quot;</td>
<td>Specifies the password of the DD Boost user.</td>
</tr>
</tbody>
</table>

### Table 4  Additional restore command options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-C</td>
<td>Relocates the database files (.mdf and .ldf) to a different folder.</td>
</tr>
<tr>
<td>-f</td>
<td>Overwrites the existing database with the current database that you restore, if the names of both the databases are same.</td>
</tr>
<tr>
<td>-j</td>
<td>Performs a database consistency check between the SQL Server backed up data and the SQL Server restored data.</td>
</tr>
<tr>
<td>-q</td>
<td>Displays <code>ddbmsqlrc</code> messages in the quiet mode, that is, the option provides minimal information about the progress of the restore operation including error messages.</td>
</tr>
<tr>
<td>-k</td>
<td>Performs checksum before restoring the data from the device.</td>
</tr>
<tr>
<td>-u</td>
<td>Performs checksum and continues the operation even in case of errors.</td>
</tr>
</tbody>
</table>
The path includes one of the following parameters:

- **Default instance or d-path:** MSSQL: or [MSSQL:<s-path>][MSSQL:<s-path>][...]
  
  Where, s-path is the database

  **Example:**
  
  ddbmsqlrc -c NMMDA241.heroines.local -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.77.27" -a "NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/heroines" "MSSQL:database1" "MSSQL:database2"

- **Named instance or i-path:** MSSQL$<Instance_Name>: or MSSQL$<Instance_Name><s-path>[...]

  Where, s-path is the database

  **Example:**
  
  ddbmsqlrc -c NMMDA241.heroines.local -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.77.27" -a "NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/heroines" "MSSQL$instance1:database1" "MSSQL$instance1:database2"

**Note:** Use the **Script** option in the **Restore** page in the Microsoft application agent for SQL Server GUI to generate the CLI equivalent script that you can use to perform tasks such as automation, and so on.
Using the T-SQL script to perform the Microsoft application agent for SQL Server restores

The Microsoft application agent for SQL Server enables you to generate SQL-CLR script that you can use to perform the Microsoft application agent for SQL Server restores in the context of a SQL Server similar to any T-SQL command or script.

The SQL-CLR restore command is `emc_run_restore`. You must provide a VARCHAR type parameters to this command. This command uses the same Microsoft application agent for SQL Server restore CLI options. So, you must have a detailed knowledge of the Microsoft application agent for SQL Server restore CLI options to flawlessly use the `emc_run_restore` command. “Using CLI to perform the Microsoft application agent for SQL Server restores” on page 76 provides information about the Microsoft application agent for SQL Server restore CLI options.

Sample T-SQL restore script

```sql
USE [master]
GO
DECLARE @returnCode int
EXEC   @returnCode = dbo.emc_run_restore ' -c win8sqlsp.sharepoint.com
        -f -t "02/03/2015 04:04:36 AM" -S normal -a "NSR_DFA_SI=TRUE" -a
        "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=10.31.192.10" -a
        "NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/ddsub7" -d
        "MSSQL:dbtest" "MSSQL:db100"

IF @returnCode <> 0
BEGIN
RAISERROR ('Fail!', 16, 1)
END
```

You can use any SQL Server standard interfaces such as, SSMS (Query window) and OSQL command prompt to run the SQL-CLR scripts.

"Recommendations for better performance of backups and restores" on page 13 provides guidelines for better performance of the Microsoft application agent for SQL Server restores by using the SQL-CLR scripts.

Performing the Microsoft application agent for SQL Server push restores by using the T-SQL script

To restore either all the databases or only the required databases of a SQL Server instance from a Data Domain device to a destination host by using a different host, perform the following steps:

1. Start SSMS on the different host that you use to perform the restore.
2. By using SSMS, connect to the destination SQL Server instance on the destination host to restore from the Data Domain device.
3. In the SSMS window on the different host, click New Query.
4. In the **New Query** window, run the T-SQL script to perform the restore.

You can either generate the T-SQL script by using the Microsoft application agent GUI on the destination host and copy it to the **New Query** window on the different host or write the T-SQL script in the **New Query** window. “Using SSMS to perform the Microsoft application agent for SQL Server restores” on page 68 and “Using the T-SQL script to perform the Microsoft application agent for SQL Server restores” on page 79 provide information.

**Sample T-SQL push restore script**

```sql
USE [master]
GO
DECLARE @returnCode int
EXEC   @returnCode = dbo.emc_run_restore ' -c clust-sql-01.contoso.com
-A clust-sql-02.contoso.com -f -t '06/22/2015 02:38:48 PM" -S normal
-a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=nmmdctwo.sp2010.com" -a
"NSR_DFA_SI_DD_USER=ost" -a "NSR_DFA_SI_DEVICE_PATH=/ddbmav2b75" -d
"MSSQL$Inst1:CLUST-SQL-02-DB02" "MSSQL$Inst1:CLUST-SQL-02-DB01"
IF @returnCode <> 0
BEGIN
RAISERROR ('Fail!', 16, 1)
END
```

Where:
- **clust-sql-01.contoso.com** is the source host from which the backup was performed.
- **Inst1** is the destination SQL Server instance, to which you must connect from the different host to perform the restore.
- **CLUST-SQL-02-DB02** is the destination database.
- **CLUST-SQL-02-DB01** is the database that was backed up from the source host.

**Performing the Microsoft application agent for SQL Server redirected restores**

You can perform the Microsoft application agent for SQL Server redirected restores by using either SSMS or CLI.

**Using SSMS to perform the Microsoft application agent for SQL Server redirected restores**

The procedure to perform the Microsoft application agent for SQL Server redirected restores is the same as the procedure that “Using SSMS to perform the Microsoft application agent for SQL Server restores” on page 68 describes except for the following change:

In step 4a, when you specify the **SQL Server host** field, select a SQL Server host that is not the current host.
Performing the Microsoft application agent for SQL Server redirected restores

When you want to restore a database from a Data Domain device to a different host that is not the source host from which you backed up the database, run a command that is similar to the following command on the target host:

```bash
dbmsqlrc.exe -c SERVER1 -f -t "12/24/2013 11:58:52" -S normal -a "NSR_DFA_SI=TRUE" -a "NSR_DFA_SI_USE_DD=TRUE" -a "NSR_DFA_SI_DD_HOST=DDHost" -a "NSR_DFA_SI_DD_USER=BoostUser1" -a "NSR_DFA_SI_DEVICE_PATH=/StorageUnit1" -d "MSSQL$MSSQL2:DB1" "MSSQL$MSSQL1:DB1"
```

where:

- **SERVER1** is the source host from which the backup was performed.
- **12/24/2013 11:58:52** is the point-in-time to restore which is optional.
- **normal** is the type of the restore which is optional.
- **/StorageUnit1** is the name of the storage unit from which you backed up the database.
- **MSSQL2** and **DB1** in **MSSQL$MSSQL2:DB1** are the names of the SQL instance and the database respectively on the target host to which you want to restore the database.
- **MSSQL1** and **DB1** in **MSSQL$MSSQL1:DB1** are respectively the names of the SQL instance and the database that were backed up from the source host.

**Note:** To customize and use the redirected restore command syntax as required, use the necessary key options and additional options that “Using CLI to perform the Microsoft application agent for SQL Server restores” on page 76 describes.

Alternatively, use the **Script** option in the **Restore** page in the Microsoft application agent for SQL Server GUI to generate the CLI equivalent script on the actual host, modify the script according to the new host requirements, and run the script on the new host to perform the redirected restores.

**Note:** Use the same Data Domain device and DD Boost username that you used to back up the database to perform a redirected restore.
Performing the Microsoft application agent for SQL Server disaster recovery

Perform the following steps to perform the Microsoft application agent for SQL Server disaster recovery:

1. Create a target Windows system with the same name as the source host name.
2. Install a SQL Server instance with the same name as the source instance name.
3. Install the Microsoft application agent for SQL Server on the target host.
4. Browse the backups of the source instance by selecting the appropriate storage unit.
5. Restore the system databases such as, master, model, msdb, and so on to the target instance.
6. Restore all the user databases to the target instance.
APPENDIX A
Deleting the Microsoft Application Agent for SQL Server Expired Backups

This appendix includes the following sections:

- Prerequisites ........................................................................................................... 84
- Using the expiry tool to delete the Microsoft application agent for SQL Server expired backups .................................................................................................................. 85
Deleting the Microsoft Application Agent for SQL Server Expired Backups

Prerequisites

Ensure that you meet the following prerequisites before you use the expiry tool to delete the Microsoft application agent for SQL Server expired backups:

- You have maintained a log of the full and incremental backups that you performed. You have manually checked whether the expired backup to delete has dependent non-expired backups. You must not delete an expired backup if the backup has dependent non-expired backups. The expiry tool does not check for the dependent backups.

- You have performed the following steps:
     For example, create the C:\ddconfig.cfg file with the following contents:
      
      ```
      DEVICE_PATH=/artitest
      DDBOOST_USER=ost
      DEVICE_HOST=10.31.192.10
      ```
     
      **Note:** For the DEVICE_HOST variable, specify either the same hostname or the same IP address that you specified in the DataDomain Server field when you added or configured the Data Domain server. The hostname must be in the FQDN format. For example, test1.contoso.com.
      
      ```
      CLIENT=MW2K8X64SQL2.nmmdev.com
      DEBUG_LEVEL=0
      LOCKBOX_PATH="C:\Program Files\EMC DD Boost Modules\DDBMA v2 esc\Lockbox"
      ```
  2. Create the lockbox folder in the C:\Program Files\EMC DD Boost Modules\DDBMA\config folder.
  3. Copy all the files from the config folder to the lockbox folder.
     Perform this step whenever the lockbox settings have changed.
Using the expiry tool to delete the Microsoft application agent for SQL Server expired backups

The expiry tool enables you to perform the following tasks:

- **Viewing the backups**
  To view the backups that are within a specific save time range, run the following command:
  
  ddbmexptool [-v -b <start_time> -e <end_time> -a <saveset_name>] -n <application_id> -z <config_file_path>  
  
  **Note:** If you do not specify the -b and -e options, the command displays all the backups.

- **Deleting the backups**
  To delete the backups that are within a specific save time range and regardless of the expiry date, run the following command:
  
  ddbmexptool -d [-v -b <start_time> -e <end_time> -a <saveset_name>] -n <application_id> -z <config_file_path>  
  
  **Note:** Do not run the command without the -b and -e options unless you want delete all the backups. If you do not specify the -b and -e options, the command deletes all the backups.

- **Deleting the expired backups**
  To delete the expired backups that are within a specific save time range, run the following command:
  
  ddbmexptool -k [-v -b <start_time> -e <end_time> -a <saveset_name>] -n <application_id> -z <config_file_path>  
  
  **Note:** If you do not specify the -b and -e options, the command deletes all the expired backups.

**Note:** The deletion commands do not check for the dependencies of the backups to delete.
Deleting the Microsoft Application Agent for SQL Server Expired Backups

Command options and description

The following table lists the command options and their description:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>Deletes the backups that are within a specific save time range.</td>
</tr>
<tr>
<td>-k</td>
<td>Deletes the expired backups that are within a specific save time range.</td>
</tr>
<tr>
<td>-v</td>
<td>Displays the verbose output on the console.</td>
</tr>
</tbody>
</table>
| -b     | Specifies the lower boundary of the save time of the backup.  
        | **Note:** If you do not specify the -d and -k options, -b specifies the lower boundary of the save time by default.  
        | Specify the time in either the **Hr (24 hour format):Min:Sec Month DD, YYYY format** or the relative time format. The supported relative time formats are **now, <n> days ago, <n> months ago, <n> years ago, and -<n> days.**  
        | If you do not specify the -b option, the command considers **1/1/1970** as the lower boundary. |
| -e     | Specifies the upper boundary of the save time of the backup.  
        | **Note:** If you do not specify the -d and -k options, -e specifies the upper boundary of the save time by default.  
        | Specify the time in either the **Hr (24 hour format):Min:Sec Month DD, YYYY format** or the relative time format. The supported relative time formats are **now, <n> days ago, <n> months ago, <n> years ago, and -<n> days.**  
        | If you do not specify the -e option, the command considers the current time as the upper boundary. |
| -a     | Specifies a filter on the save set name for display and deletes both. |
| -n     | Specifies the application type.  
        | **Note:** For SQL Server, application_id must be mssql. |
| -z     | Specifies the configuration filepath. |
| -Y     | Indicates your approval (yes) to delete the expired backups.  
        | If you specify the -Y option, the message that asks for your approval to delete the expired backups does not appear. |