

Teradata® Data Mover

Installation, Configuration, and Upgrade Guide
for Customers

Release 17.20




2022-09-26

Copyright and Trademarks

Copyright © 2015 - 2022 by Teradata. All Rights Reserved.

All copyrights and trademarks used in Teradata documentation are the property of their respective owners. For more information, see [Trademark Information](#).

Product Safety

Safety type	Description
 NOTICE	Indicates a situation which, if not avoided, could result in damage to property, such as to equipment or data, but not related to personal injury.
 CAUTION	Indicates a hazardous situation which, if not avoided, could result in minor or moderate personal injury.
 WARNING	Indicates a hazardous situation which, if not avoided, could result in death or serious personal injury.

Third-Party Materials

Non-Teradata (i.e., third-party) sites, documents or communications ("Third-party Materials") may be accessed or accessible (e.g., linked or posted) in or in connection with a Teradata site, document or communication. Such Third-party Materials are provided for your convenience only and do not imply any endorsement of any third party by Teradata or any endorsement of Teradata by such third party. Teradata is not responsible for the accuracy of any content contained within such Third-party Materials, which are provided on an "AS IS" basis by Teradata. Such third party is solely and directly responsible for its sites, documents and communications and any harm they may cause you or others.

Warranty Disclaimer

Except as may be provided in a separate written agreement with Teradata or required by applicable laws, all designs, specifications, statements, information, recommendations and content (collectively, "content") available from the Teradata Documentation website or contained in Teradata information products is presented "as is" and without any express or implied warranties, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or noninfringement, which are hereby disclaimed. In no event shall Teradata corporation, its suppliers or partners be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of content, even if advised of the possibility of such damage.

The Content available from the Teradata Documentation website or contained in Teradata information products may contain references or cross-references to features, functions, products, or services that are not announced or available in your country. Such references do not imply that Teradata Corporation intends to announce such features, functions, products, or services in your country. Please consult your local Teradata Corporation representative for those features, functions, products, or services available in your country.

The Content available from the Teradata Documentation website or contained in Teradata information products may be changed or updated by Teradata at any time without notice. Teradata may also make changes in the products or services described in the Content at any time without notice.

The Content is subject to change without notice. Users are solely responsible for their application of the Content. The Content does not constitute the technical or other professional advice of Teradata, its suppliers or partners. Users should consult their own technical advisors before implementing any Content. Results may vary depending on factors not tested by Teradata.

Machine-Assisted Translation

Certain materials on this website have been translated using machine-assisted translation software/tools. Machine-assisted translations of any materials into languages other than English are intended solely as a convenience to the non-English-reading users and are not legally binding. Anybody relying on such information does so at his or her own risk. No automated translation is perfect nor is it intended to replace human translators. Teradata does not make any promises, assurances, or guarantees as to the accuracy of the machine-assisted translations provided. Teradata accepts no responsibility and shall not be liable for any damage or issues that may result from using such translations. Users are reminded to use the English contents.

Feedback

To maintain the quality of our products and services, e-mail your comments on the accuracy, clarity, organization, and value of this document to: docs@teradata.com.

Any comments or materials (collectively referred to as "Feedback") sent to Teradata Corporation will be deemed nonconfidential. Without any payment or other obligation of any kind and without any restriction of any kind, Teradata and its affiliates are hereby free to (1) reproduce, distribute, provide access to, publish, transmit, publicly display, publicly perform, and create derivative works of, the Feedback, (2) use any ideas, concepts, know-how, and techniques contained in such Feedback for any purpose whatsoever, including developing, manufacturing, and marketing products and services incorporating the Feedback, and (3) authorize others to do any or all of the above.

Confidential Information

Confidential Information means any and all confidential knowledge, data or information of Teradata, including, but not limited to, copyrights, patent rights, trade secret rights, trademark rights and all other intellectual property rights of any sort.

The Content available from the Teradata Documentation website or contained in Teradata information products may include Confidential Information and as such, the use of such Content is subject to the non-use and confidentiality obligations and protections of a non-disclosure agreement or other such agreements to protect Confidential Information that you have executed with Teradata.

Contents

Chapter 1: Overview	6
Welcome to Teradata Data Mover Installation, Configuration, and Upgrade Guide for Customers	6
Dependencies	7
Data Mover Network Diagram	8
Data Mover Best Practices	18
Part I: Installing and Configuring Software	19
Chapter 2: Configuring the Environment	20
Configuring the Data Mover Daemon	20
Configuring the Data Mover Agent	27
Configuring the Data Mover Command-Line Interface	31
Configuring the Data Mover REST Service	34
Configuring the Cloud Staging Copy REST Service	36
DSA Configurations	37
Cloud Staging Copy Service with Data Mover	41
High Availability Overview	44
Configuring Data Mover to Use Teradata Ecosystem Manager	69
Configuring Multiple Multi-Purpose Servers	70
Configuring Data Mover to Log to Server Management	70
Configuring Data Mover Multi-Purpose Server to Increase Network Throughput	71
Data Mover Log Files	73
Data Mover Properties Files Preserved During Upgrades	74
DSA Setup for New Teradata Systems	74
Chapter 3: Deploying Data Mover on VMware	96
About Data Mover on VMware	96
Downloading Data Mover for VMware	97
Deploying Data Mover on VMware	97
Chapter 4: Administrative Tasks	99
Data Mover Components Script	99
Changing DBC, DATAMOVER, and DSA Passwords on the Data Mover Server	99
Changing POSTGRES, DATAMOVER and DSA Passwords on the Data Mover Server	102
Creating a Diagnostic Bundle for Support	105
Migrating Existing ARC Jobs to DSA	107
Migrating the Teradata Repository to the Postgres Repository	107
Configuring ActiveMQ on Remote Agents	107

Part II: Upgrading Software	109
Chapter 5: Upgrading Software	110
Upgrading Data Mover Software	110
Creating an Incident	110
Upgrading the Data Mover Command-Line Interface on Non-Teradata Servers	110
Upgrading the Data Mover Agent on a Linux Teradata Server	113
Installing or Upgrading the Data Mover Portlet	113
Appendix A: Additional Information	115

Overview

Welcome to Teradata Data Mover Installation, Configuration, and Upgrade Guide for Customers

Using Teradata Data Mover Installation, Configuration, and Upgrade Guide for Customers

This guide describes the procedure to install and upgrade the Data Mover portlet, and system requirements, configuration, and deployment of the Teradata® Data Mover components. The administrative tasks are also available in the guide.

The contents are for system administrators, database administrators and relational database developers, customers, and Teradata customer support. Only the root users can install, configure, and perform administrative tasks on Data Mover components.

Why Would I Use this Content?

This document has the following information related to Data Mover, which helps you to start the Data Mover setup:

- Required open ports on the Data Mover server
- System requirements
- Network diagrams
- Configuration of different components
- Deployment on VMware
- Procedures to perform administrative tasks
- Upgrade and installation procedures

How Do I Use this Content?

Refer to this document as per your role and requirements:

- If you are a root user, refer to the installation, configuration, and administrative task sections to work with Data Mover components.
- If you are a beginner, go through the port information and network diagrams to understand the Data Mover architecture.
- If you are familiar with vSphere, vCenter, ESXi terminology and understand switches, datastores, VM templates, .ova, .ovf, and other database components, use this content to deploy Data Mover on VMware.

How Do I Get Started?

Start with the [Overview](#) section to understand the product and prepare your system for Data Mover. Go through the [Deploying Data Mover on VMware](#) section to get details on VMware deployment. Refer to the respective sections for further requirements such as:

- [Upgrading Software](#): Install and upgrade Data Mover portlet and software.
- [Configuring the Environment](#): Configure Data Mover components.
- [Deploying Data Mover on VMware](#): Deploy Data Mover on VMware.
- [Administrative Tasks](#): Administrative tasks for administrators.

References to Other Relevant Software

- To back up the DSA repository, refer to *Teradata® DSA User Guide*, B035-3150.
- To install the DSA, refer to *Teradata® Data Stream Utility Installation, Configuration, and Upgrade Guide*, B035-3153.

Dependencies

For Teradata Data Mover dependencies and compatibility information, see the [Teradata® Data Mover Compatibility Matrix](#).

Required Permissions

You must be a root user to install and configure Data Mover components.

Required Open Ports on the Data Mover Server

The TCP ports listed here must be open for incoming and outgoing traffic on the Data Mover server:

Port Number	Used By
22	SSH
443/1025	TPTAPI and JDBC <ul style="list-style-type: none"> • HTTPS 443 is used when TLS 1.2 encryption is enabled. • TCP 1025 is used otherwise.
1443	HTTPS RESTful API
5432	Postgres Repository
443/9981	Server Management <ul style="list-style-type: none"> • Port 443 is used for CMIC 14.06.01 and later. • Port 9981 is used for earlier versions of CMIC.
9090	DSA RESTful API
61616	ActiveMQ

System Requirements for the Data Mover Server across All Cloud Platforms

Teradata recommends the following infrastructure for Data Mover across all cloud platforms:

Hardware and Network	Production Environment	Non-Production Environment
Disk Size (Operating System)	256 GB	256 GB
Disk Size (Data Mover Repository)	128 GB	128 GB
CPU	8 to 16	2 to 4
RAM	64 to 128 GB	8 to 16 GB
Network	10 Gbps	As per the VM size

In addition, consider the following:

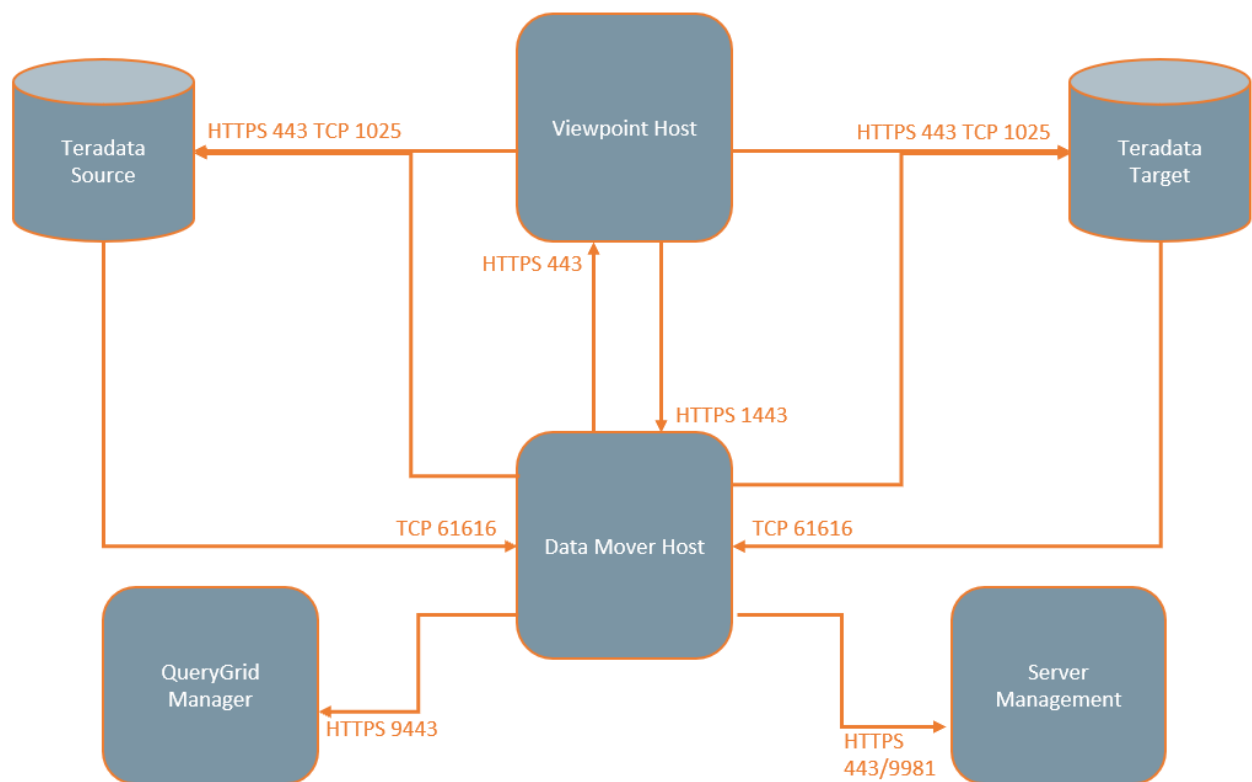
- If the environment use is minimal, then select the lowest of the recommended CPU and RAM value.
- If a non-production environment use is going to be significant, then select the system requirements recommended for production environment.
- If the network lags, then upgrade the machine accordingly. Depending on the VM Family and size, the network bandwidth may vary for every cloud service provider.

Data Mover Network Diagram

Single Data Mover Host with Viewpoint Portlet

This scenario assumes the following setup:

- Single Data Mover host with Data Mover daemon, single Data Mover agent
- Viewpoint host with Data Mover portlet
- Teradata Source/Teradata Target
- Embedded DSC on Data Mover host and ClientHandler on source or target Teradata TPA nodes

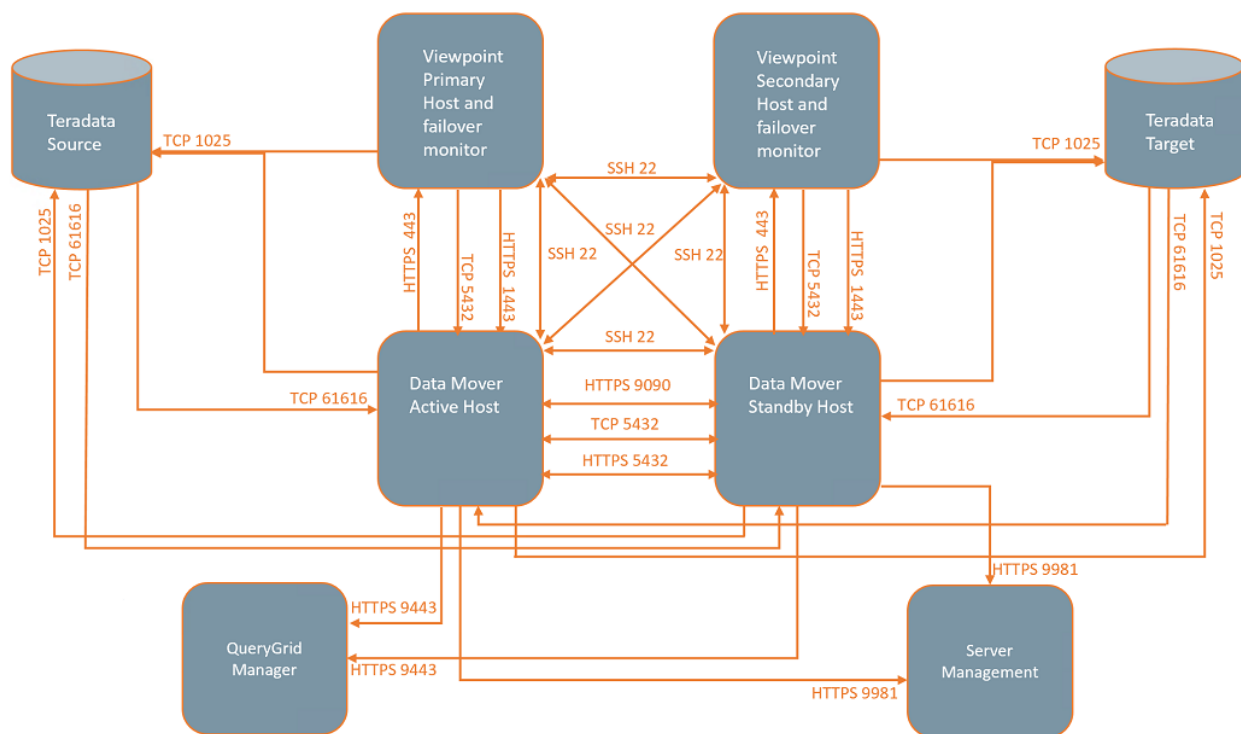


Host	Port	Connection Type	Notes
Data Mover	61616	TCP	If using embedded DSC on Data Mover host, the ClientHandler connects to Data Mover host using port 61616.
Viewpoint	443	HTTPS	If security is enabled, Data Mover connects to Viewpoint host using port 443 (80 if using HTTP).
Teradata Source/ Teradata Target	443/ 1025	HTTPS/TCP	Data Mover connects to Teradata Source/Teradata Target using HTTPS over port 443 if TLS 1.2 encryption is enabled. Otherwise Data Mover uses TCP over port 1025. Viewpoint Data Mover portlet connects to Teradata Source/Teradata Target using port 1025.
Data Mover	1443	HTTPS	The Data Mover Viewpoint portlet connects to the Data Mover host using Data Mover REST API port 1443.
Server Management	443/ 9981	HTTPS	Data Mover connects to Server Management using HTTPS using port 443 when using CMIC version 14.06.01 or later. Earlier versions of CMIC use port 9981.
QueryGrid Manager	9443	HTTPS	Data Mover connects to QueryGrid Manager for status on T2T jobs using HTTPS using port 9443.

HA Data Mover Cluster

This scenario assumes the following setup:

- Active Data Mover host with Data Mover daemon, single Data Mover agent
- Standby Data Mover host with offline Data Mover daemon, online single Data Mover agent
- Viewpoint cluster with each host having Data Mover portlet and Data Mover failover monitor
- Embedded DSC on both active and standby Data Mover host (only DSC on active Data Mover host online)
- ClientHandler on Teradata Source/Teradata Target TPA nodes
- Teradata Source/Teradata Target



If Data Mover failover occurs, the active and standby Data Mover host switches roles. Therefore, when setting up ports, configure both active and standby Data Mover host in the same manner so that either can function as the active Data Mover host.

Host	Port	Connection Type	Notes
Active Data Mover Host	61616	TCP	If using embedded DSC on Data Mover host, the ClientHandler connects to Data Mover host using port 61616.
Active Data Mover Host	22	SSH	The Data Mover failover monitor running on Viewpoint hosts connects to Data Mover host using port 22.

Host	Port	Connection Type	Notes
			The Data Mover cluster command running on standby Data Mover host connects to Data Mover host using port 22.
Active Data Mover Host	5432	TCP	The standby postgres sync running on standby Data Mover host connects to postgres logical sync service using port 5432. The Data Mover failover monitor runs JDBC query to verify repo status by connecting to port 5432.
Active Data Mover Host	443	HTTPS	The Data Mover Viewpoint portlet connects to the Data Mover host using Data Mover REST API port 443.
Active Data Mover Host	9090	HTTPS	Data Mover Agent on secondary Data Mover host connects to DSA REST running on primary Data Mover host using port 9090.
Standby Data Mover Host	61616	TCP	If using embedded DSC on Data Mover host, the Clienthandler connects to Data Mover host using port 61616.
Standby Data Mover Host	22	SSH	The Data Mover failover monitor running on Viewpoint hosts connects to Data Mover host using port 22. The Data Mover cluster command running on standby Data Mover host connects to Data Mover host using port 22.
Standby Data Mover Host	443	HTTPS	The Data Mover Viewpoint portlet connects to Data Mover host using DM REST API port 443.
Standby Data Mover Host	5432	TCP	The Data Mover failover monitor runs JDBC query to verify repo status by connecting to port 5432.
Standby Data Mover Host	9090	HTTPS	If failover occurs, external Data Mover Agent connects to DSA REST running on secondary Data Mover host using port 9090.
Primary Viewpoint Host	443	HTTPS	If security is enabled, Data Mover connects to Viewpoint host using port 443 (80 if using HTTP).
Primary Viewpoint Host	22	SSH	The Data Mover failover monitor running on secondary Viewpoint host connects to primary Viewpoint host using port 22. The Data Mover cluster command running on either active or standby Data Mover host connects to Viewpoint host using port 22.
Primary Viewpoint Host	5432	TCP	The Data Mover failover monitor connects to Active Data Mover host to check whether repository is running using JDBC connection.
Secondary Viewpoint Host	443	HTTPS	If security is enabled, Data Mover connects to Viewpoint host using port 443 (80 if using HTTP).

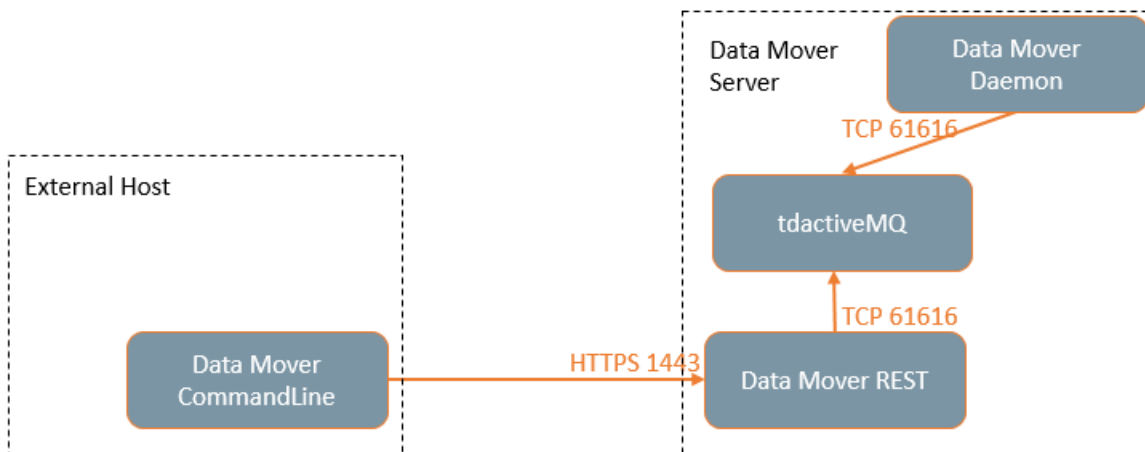
Host	Port	Connection Type	Notes
Secondary Viewpoint Host	22	SSH	The Data Mover failover monitor running on secondary Viewpoint host connects to primary Viewpoint host using port 22. The Data Mover cluster command running on either active or standby Data Mover host connects to Viewpoint host using port 22.
Secondary Viewpoint Host	5432	TCP	The Data Mover failover monitor connects to standby Data Mover host to check whether repository is running using JDBC connection.
Teradata Source/ Teradata Target	443/ 1025	HTTPS/TCP	Data Mover connects to Teradata Source/Teradata Target using HTTPS over port 443 if TLS 1.2 encryption is enabled. Otherwise TCP over port 1025 is used. Viewpoint Data Mover portlet connects to Teradata Source/Teradata Target using port 1025.
Server Management	443/ 9981	HTTPS	Data Mover connects to Server Management using HTTPS using port 443/9981. <ul style="list-style-type: none"> Port 443: used if CMIC is 14.06.01 or later. Earlier versions of CMIC use port 9981.
QueryGrid Manager	9443	HTTPS	Data Mover connects to QueryGrid Manager for status on T2T jobs using HTTPS using port 9443.

Additional Configurations

Data Mover Command Line Running on External Client Host

This scenario assumes the following setup:

- Data Mover CommandLine is installed on external client host and configured to connect to Data Mover host
- Host information is added to the `accept.host.list` property. For more information on the Accept Host List, refer to the RESTful API section of the *Teradata® Data Mover User Guide*, B035-4101



Host	Port	Connection Type	Notes
Data Mover	1443	HTTPS	The Data Mover CommandLine connects to the Data Mover host (active or standby) using Data Mover REST API port 1443.
Data Mover	61616	TCP	Data Mover REST and Data Mover daemon connects to tdactiveMQ using port 61616.

Client Calling Data Mover REST API from External Host

This scenario assumes the following setup:

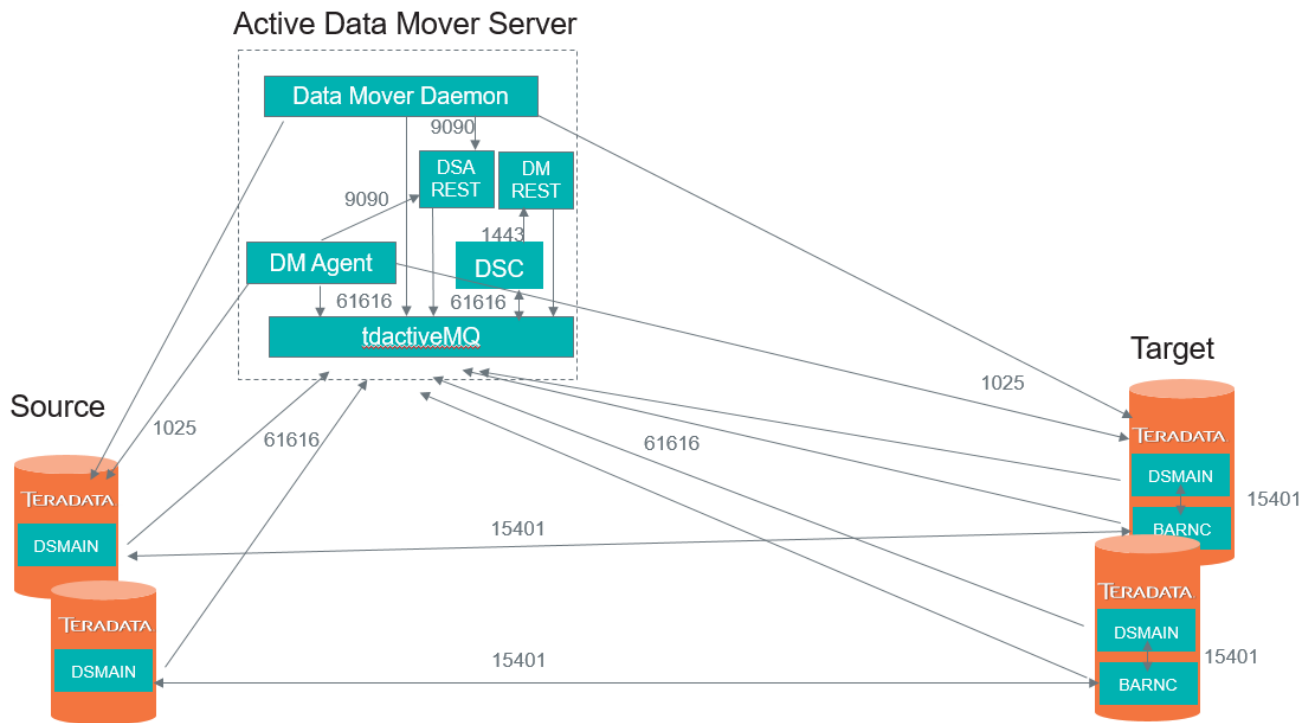
- The client is making REST calls to the Data Mover REST API from an external host
- Host information is added to the `accept.host.list` property. For more information on the Accept Host List, refer to the RESTful API section of the *Teradata® Data Mover User Guide*, B035-4101

Host	Port	Connection Type	Notes
Data Mover	1443	HTTPS	The client connects to the Data Mover host using port 1443.

Data Mover Using Embedded DSC

This scenario assumes the following setup:

- Data Mover is configured to use embedded DSC running on Data Mover host



Host	Port	Connection Type	Notes
Data Mover	61616	TCP	DSA NC (network client), DSMain connecting to tdactiveMQ to communicate with DSC.
Data Mover	9090	HTTPS	Internal connection in this configuration. Data Mover connect to DSA REST on same server using port 9090.
Data Mover	1443	HTTPS	Internal connection in this configuration. DSC connects to DM REST on same server using port 1443.
Teradata TPA Node	15401	TCP	DSMain connecting to DSA NC (assuming NC is installed on TPA node). DSA NC connecting to other DSA NC on other TPA nodes (assuming NC installed on TPA node).
Teradata TPA Node	443/ 1025	HTTPS/TCP	Data Mover connects to Teradata Source/Teradata Target using HTTPS over port 443 if TLS 1.2 encryption is enabled. Otherwise TCP over port 1025 is used.

Data Mover Using External DSC

This scenario assumes the following setup:

- Data Mover is configured to use external DSC running on an external host

Host	Port	Connection Type	Notes
Data Mover	1443	HTTPS	The external DSC connects to Data Mover host using port 1443.
DSC	9090	HTTPS	Data Mover connects to the DSC host using port 9090.
DSC	61616	TCP	DSA NC, DSMain connecting to tdactiveMQ to communicate with DSC.
Teradata TPA Node	15401	TCP	DSMain connecting to DSA NC (assuming NC installed on TPA node). DSA NC connecting to other DSA NC on other TPA nodes (assuming NC installed on TPA node).
Teradata TPA Node	443/1025	HTTPS/TCP	Data Mover connects to Teradata Source/Teradata Target using HTTPS over port 443 if TLS 1.2 encryption is enabled. Otherwise TCP over port 1025 is used.

Data Mover Using QueryGrid 2.x+

This scenario assumes the following setup:

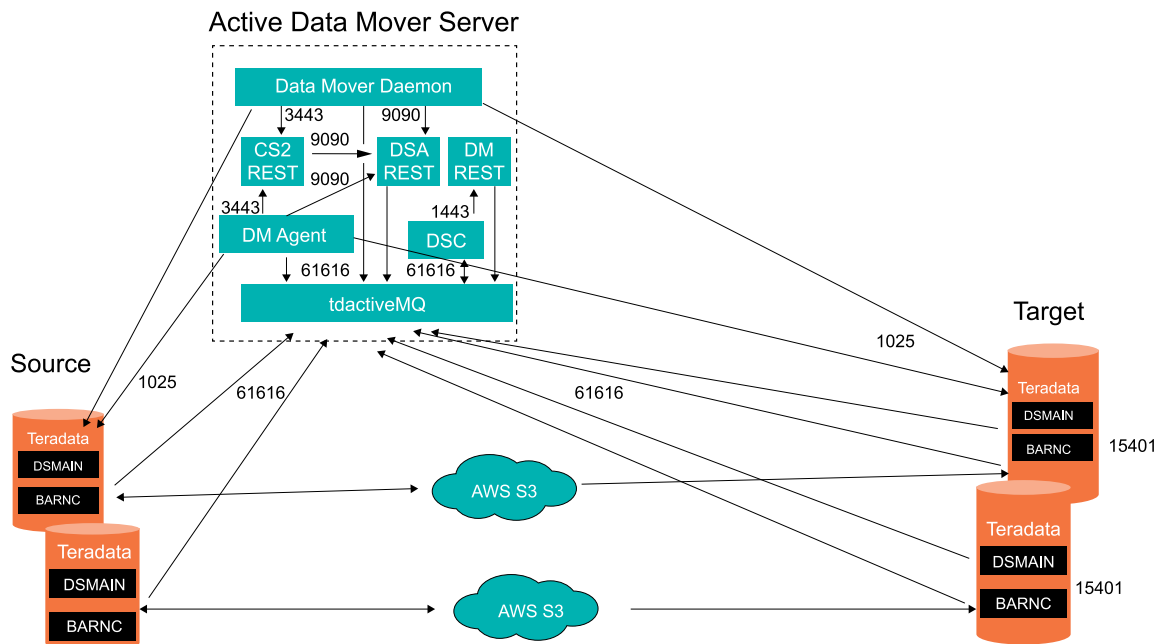
- Data Mover is configured to use QueryGrid 2.x and therefore is configured to call QueryGrid Manager on an external host

Host	Port	Connection Type	Notes
QueryGrid Manager	9443	HTTPS	Data Mover connects to the QueryGrid Manager host using port 9443.

Data Mover Using Cloud Staging Copy Service (CS2) with Embedded DSC

This scenario assumes the following setup:

- Data Mover is configured to use Cloud Staging Copy Service (CS2) with the embedded DSC running on Data Mover host.



Host	Port	Connection Type	Notes
Data Mover	61616	TCP	DSA NC (network client), DSMain connecting to tdactiveMQ to communicate with DSC.
Data Mover	9090	HTTPS	Internal connection in this configuration. Data Mover and CS2 connect to DSA REST on same server using port 9090.
Data Mover	1443	HTTPS	Internal connection in this configuration. DSC connects to DM REST on same server using port 1443.
Teradata TPA Node	15401	TCP	DSMain connecting to DSA NC (assuming NC is installed on TPA node). DSA NC connecting to other DSA NC on other TPA nodes (assuming NC installed on TPA node).
Teradata TPA Node	443/ 1025	HTTPS/TCP	Data Mover connects to Teradata Source/Teradata Target using HTTPS over port 443 if TLS 1.2 encryption is enabled. Otherwise TCP over port 1025 is used.
Data Mover	3443	HTTPS	Internal connection in this configuration. Data Mover communicate with CS2 REST.

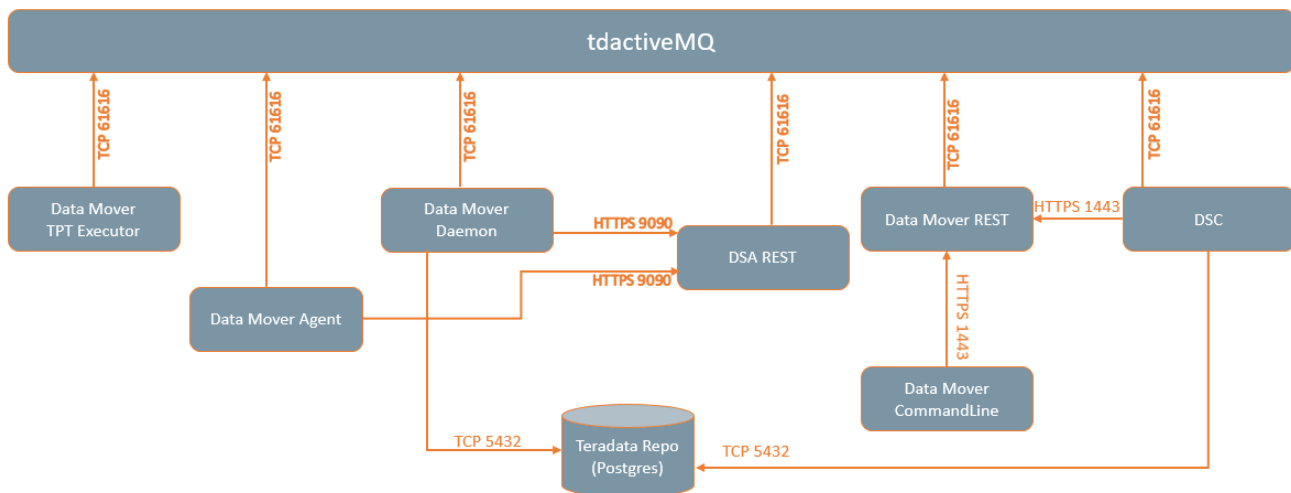
Data Mover and Single Sign On

If an IdP is integrated with Viewpoint and Viewpoint sends a JWT to Data Mover, then Data Mover connects to the IdP REST API. The default https port of IdP is 443, but if that changes, then Data Mover must have access to the different port.

Internal Connections on Data Mover Host

This scenario assumes the following setup:

- Data Mover host with Data Mover daemon, single Data Mover agent
- Embedded DSC

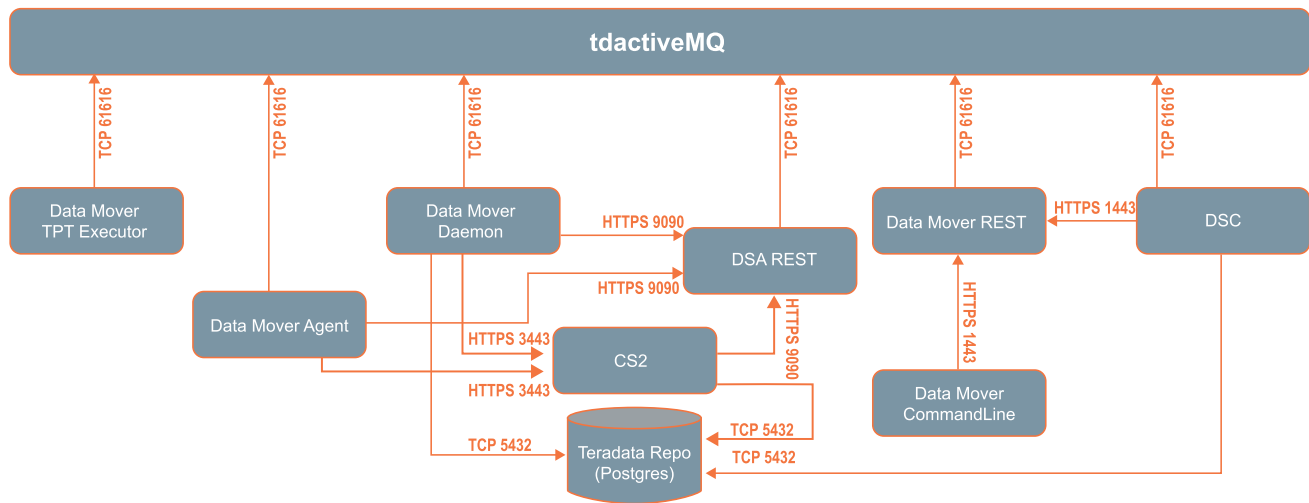


Host	Port	Connection Type	Notes
Data Mover	61616	TCP	The Data Mover daemon, Data Mover agent, Data Mover TPT Executor, Data Mover REST, DSA REST, and DSC all connect to tdactiveMQ using port 61616.
Data Mover	5432	TCP	The Data Mover daemon and DSC connect to the local Teradata repository using port 5432.
Data Mover	1443	HTTPS	The DSC and Data Mover CommandLine connects to Data Mover REST using port 1443.
Data Mover	9090	HTTPS	The Data Mover daemon and Data Mover agent connect to DSA REST using port 9090.

Internal Connections on Data Mover with Cloud Staging Copy Service (CS2)

This scenario assumes the following setup:

- Data Mover host with Data Mover daemon, single Data Mover agent
- Embedded DSC
- Cloud Staging Copy Service (CS2) with Data Mover



Host	Port	Connection Type	Notes
Data Mover	61616	TCP	The Data Mover daemon, Data Mover agent, Data Mover TPT Executor, Data Mover REST, DSA REST, and DSC all connect to tdactiveMQ using port 61616.
Data Mover	5432	TCP	The Data Mover daemon and DSC connect to the local Teradata repository using port 5432.
Data Mover	1443	HTTPS	The DSC and Data Mover CommandLine connects to Data Mover REST using port 1443.
Data Mover	9090	HTTPS	The Data Mover daemon, Data Mover agent, and CS2 connect to DSA REST using port 9090.
Data Mover	3443	HTTPS	The Data Mover daemon and Data Mover agent connect to CS2 using port 9090.

Data Mover Best Practices

A comprehensive implementation and configuration best practices guide is available for Data Mover. This guide includes networking information to help you understand and resolve a variety of performance issues, including test and validation procedures for suggested changes.

The Data Mover Best Practices Guide is available online and for download at [Teradata® Data Mover Best Practices Guide](#).

Installing and Configuring Software

Configuring the Environment

Configuring the Data Mover Daemon

1. Edit the `daemon.properties` file and restart the Data Mover daemon to implement the changes. For properties that can be set dynamically, the changes take effect one minute after the updated `daemon.properties` file is saved. There is no need to restart the daemon service if you are only updating dynamic properties.
2. Use the `list_configuration` and `save_configuration` commands to modify the other Data Mover properties.

The `daemon.properties` File

Property	Description	Default Value
<code>broker.port=port</code>	The port number of the machine where the Java Message Service (JMS) message broker is listening.	61616
<code>broker.url=url</code>	The hostname or IP address of the machine running the Java Message Service (JMS) message broker.	localhost
<code>cluster.enabled=setting for cluster</code>	When set to <code>True</code> , establishes a connection to a standby Java Message Service (JMS) broker in case the active JMS broker fails.	False
<code>viewpoint.url</code>	The hostname or IP address for the Viewpoint Authentication server. Example: <code>viewpoint.url=https://localhost</code>	<code>https://localhost</code>
<code>viewpoint.port</code>	The port number for the Viewpoint Authentication server. Example: <code>viewpoint.port=443</code>	443
<code>tvi.useLogger=setting for TVI messages</code>	The Server Management logger can be set to <code>true</code> or <code>false</code> . If set to <code>true</code> , fatal error messages are sent to Server Management. Dynamic property. ¹	True
<code>jobExecutionCoordinator.maxConcurrentJobs=maximum number of jobs</code>	The maximum number of jobs allowed to run on the daemon at the same time. Additional jobs are placed on the queue and run when slots become available. Dynamic property. ¹	20
<code>jobExecutionCoordinator.maxQueuedJobs=maximum number of jobs allowed in queue</code>	The maximum number of jobs allowed in the job queue. Additional jobs are placed in a higher level memory queue until slots are available in the job queue. Dynamic property. ¹	20

Property	Description	Default Value
<code>querygrid.manager.urls=url</code>	The hostname and IP address for the QueryGrid Manager servers. Supports up to two URLs, separated by commas. Example: <code>querygrid.manager.urls=https://host1:9443,https://host2:9443</code>	9443
<code>rootLogger.level</code>	The six levels of logging, TRACE DEBUG INFO WARN ERROR FATAL. From trace level to application error. LOG_LEVEL can be updated dynamically, but not logfile. ¹	INFO
<code>appender.rolling.type</code>	The six levels of logging, TRACE DEBUG INFO WARN ERROR FATAL. From trace level to application error. LOG_LEVEL can be updated dynamically.	RollingFile
<code>appender.rolling.name</code>	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
<code>appender.rolling.layout.type</code>	Do not edit. This is an internal setting for logging infrastructure.	PatternLayout
<code>appender.rolling.layout.pattern</code>	Do not edit. This is an internal setting for logging infrastructure. <ul style="list-style-type: none"> d = date t = thread p = log level c = class name m = message n = platform dependent line separator 	%d [%t] %-5p %c{3}(%L) - %m%n
<code>appender.rolling.policies.type</code>	Do not edit. This is an internal setting for logging infrastructure.	Policies
<code>appender.rolling.policies.size.type</code>	Do not edit. This is an internal setting for logging infrastructure.	SizeBasedTriggeringPolicy
<code>appender.rolling.strategy.type</code>	Do not edit. This is an internal setting for logging infrastructure.	DefaultRolloverStrategy
<code>logger.rolling.name</code>	Do not edit. This is an internal setting for logging infrastructure.	<code>com.teradata.datamovement.daemon</code>
<code>logger.rolling.appenderRef.rolling.ref</code>	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
<code>appender.rolling.fileName</code>	The relative or absolute path of the log file. If changing the location, specify the absolute path of the file. For Windows, specify back slash instead of forward slash, for example, <code>C:\Program File\Teradata\Log\dmDaemon.log</code> . Dynamic property. ¹	<code>/var/opt/teradata/datamover/logs/dmDaemon.log</code>
<code>appender.rolling.filePattern</code>	Specifies logfiles pattern. Example: <code>dmDaemon.log.%i</code> saves the files as <code>dmDaemon.log.1</code> , <code>dmDaemon.log.2</code> , <code>dmDaemon.log.3</code> , and so on.	<code>/var/opt/teradata/datamover/logs/dmDaemon.log.%i</code>

Property	Description	Default Value
<code>appender.rolling.policies.size.size</code>	The maximum size of the logging file before being rolled over to backup files. Dynamic property. ¹	20MB
<code>appender.rolling.strategy.max</code>	The number of backup logging files that are created. After the maximum number of files have been created, the oldest file is deleted. Dynamic property. ¹ Example: If maximum backups = 3, three backup logs are created: <ul style="list-style-type: none"> • <code>dmDaemon.log.1</code> • <code>dmDaemon.log.2</code> • <code>dmDaemon.log.3</code> If the current <code>dmDaemon.log</code> size exceeds 20MB, it rolls to become the new <code>dmDaemon.log.1</code> and a new <code>dmDaemon.log</code> is created. The previous <code>dmDaemon.log.2</code> becomes the new <code>dmDaemon.log.3</code> . The previous <code>dmDaemon.log.3</code> is deleted.	5
<code>dm.rest.endpoint</code>	This is a hidden property used only by DSA. The Data Mover REST URL used by the daemon for the DSA utility. You can override the default value by adding the same named property in the <code>daemon.properties</code> file. Before modifying the localhost, verify the new host is listed in the <code>accept.host.list</code> property in <code>tdmrest.properties</code> file.	<code>http://localhost:1443/datamover</code>
<code>dsa.rest.endpoint</code>	This is a hidden property used only by DSA. The DSA REST URL used by the daemon for the DSA utility. You can override the default value by adding the same named property in the <code>daemon.properties</code> file.	<code>http://localhost:9090/dsa</code>
<code>is.dsc.colocate.dm</code>	This is a hidden property used only by DSA. Flag indicating if Data Mover is using the bundled DSC. You can override the default value by adding the same named property in the <code>daemon.properties</code> file.	true

If the Viewpoint Authentication server does not have HTTPS enabled, you can set the following if you want to authenticate through HTTP instead: `viewpoint.url` to `http://localhost` and `viewpoint.port` to 80.

¹For properties that can be set dynamically, the changes take effect one minute after the updated `daemon.properties` file is saved. There is no need to restart the daemon service if you are only updating dynamic properties. For example:

- If you changed the value of `rootLogger.level` from the default of `INFO`, logfile to `DEBUG`, logfile, any debug messages generated would start appearing in the log file one minute after saving the updated properties file.
- If you changed the value of `jobExecutionCoordinator.maxConcurrentJobs` from the default value of 20 to a new value of 25, the new value of 25 takes effect one minute after saving the updated `daemon.properties` file.

Configuration Properties

Property	Description	Default Value
<code>agentCollector.agentHeartbeatWaitMillis</code>	Sets the amount of time in milliseconds to wait for an agent heartbeat before assuming the agent has gone out of service.	600000
<code>blocked.job.maxAllowedLimit</code>	The maximum number of jobs that can be marked as <code>BLOCKED</code> and retried. If a job is detected as blocked when the <code>blocked.job.maxAllowedLimit</code> has already been reached, the job is added to the Job Queue. The value cannot be greater than 25% of the maximum concurrent job limit.	5
<code>blocked.job.retry.enabled</code>	When set to <code>True</code> , detects any locks on the source/target objects being moved and retries running the job after a specified interval.	False
<code>blocked.job.retry.interval</code>	Sets an interval to retry running any jobs blocked because of locks on source/target objects. Time unit can be specified as <code>HOURS</code> or <code>MINUTES</code> .	1 HOUR
<code>blocked.job.retry.maxInterval</code>	Sets the maximum interval for attempting to start any jobs blocked because of locks on source /target objects. Jobs are marked as <code>FAILED</code> after the maximum interval is exceeded the jobs are still blocked. Time unit can be specified as <code>HOURS</code> or <code>MINUTES</code> .	1 HOUR
<code>daemon.default.compareDDL.enabled</code>	Enables/disables the default <code>compareDDL</code> behavior at the daemon level.	
<code>databaseQueryService.useBaseViewsOnly</code>	Sets all data dictionary queries on Teradata source and target systems to use the base views instead of <code>X</code> or <code>VX</code> views.	True
<code>deadlock.retry.enabled</code>	When set to <code>True</code> , if an SQL query execution fails with DBS error (2631) because of a deadlock, retries executing the query after a specified interval.	False
<code>deadlock.retry.interval</code>	The interval during which to retry executing an SQL query that fails with a DBS deadlock error (2631). Time unit can be specified as <code>SECONDS</code> or <code>MINUTES</code> .	1 MINUTE
<code>deadlock.retry.maxAttempts</code>	The maximum number of attempts to retry executing an SQL query that fails with a DBS deadlock error (2631).	10

Property	Description	Default Value
<code>different.session.charsets.enabled</code>	Determines whether or not specifying different source and target session character sets in a job is allowed. Default value False means this is not allowed.	False
<code>event.table.default</code>	Default event table in which to save event details. Events are sent to this event table by default when <code>tmsm.mode</code> is either BOTH or ONLY_INTERNAL_TMSM. Individual jobs can use a different event table by using the <code>log_to_event_table</code> job definition parameter. Multiple values can be set as follows: <ul style="list-style-type: none"> • <code><value>event1</value></code> • <code><value>event2</value></code> 	NULL
<code>hanging.job.check.enabled</code>	If enabled, an internal process awakens periodically and reviews running jobs to see if any have stopped responding.	Disabled
<code>hanging.job.check.rate</code>	Rate at which to check for jobs in halt (in hours).	1 HOUR
<code>hanging.job.timeout.acquisition</code>	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout is specifically for the acquisition phase.	1 HOUR
<code>hanging.job.timeout.large.apply</code>	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the TPTAPI apply phase for a large object.	4 HOURS
<code>hanging.job.timeout.large.build</code>	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the DSA build phase for a large object.	4 HOURS
<code>hanging.job.timeout.large.initiate</code>	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the initiate phase for a large object.	4 HOURS
<code>hanging.job.timeout.medium.apply</code>	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the TPTAPI apply phase for a medium object.	2 HOURS
<code>hanging.job.timeout.medium.build</code>	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the DSA build phase for a medium object.	2 HOURS
<code>hanging.job.timeout.medium.initiate</code>	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the initiate phase for a medium object.	2 HOURS
<code>hanging.job.timeout.range.large.min</code>	Defines the minimum size (in MB, GB, TB, or default GB if the unit is not provided) for an object to be considered a large object.	10 GB
<code>hanging.job.timeout.range.small.max</code>	Defines the maximum size (in MB, GB, TB, or default MB if the unit is not provided) for an object to be considered a small object.	5 MB

Property	Description	Default Value
hanging.job.timeout.small.apply	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the TPTAPI apply phase for a small object.	1 HOUR
hanging.job.timeout.small.build	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the DSA build phase for a small object.	1 HOUR
hanging.job.timeout.small.initiate	If the progress of a new job is not reported within this period (in hours), the job is stopped. Timeout specifically for the initiate phase for a small object.	1 HOUR
job.allowCommandLineUser	When set to True, the daemon allows Command Line requests when the security level is Daemon.	False
job.databaseClientEncryption	When set to True, utilities such as DSA, JDBC, and TPTAPI initiate encrypted sessions to both the source and target database systems. Note: Performance decreases when encryption is initiated.	False
job.default.dsa.nontable	When set to true, Data Mover tries to use DSA for moving non-table objects such as macros, views and stored procedures. Note: Do not change this property while jobs are being created, edited or running.	False
job.default.foreign.server.on.target	If set to true, Data Mover tries to use foreign server on target system for T2T job.	False
job.default.queryband	Provides a set of name/value pairs to be used as the default query band for all jobs.	ApplicationName=DM; Version=16.00
job.default.queryband.enabled	Enable to use of the default query band features.	False
job.force.direction	Forces the direction of data movement from source to target system.	
job.never.target.system	Prevents certain database systems from ever being a target system in a Data Mover job.	False
job.onlineArchive	When set to True, online archiving is used for objects that merit the use of DSA. Note: Performance decreases when this setting is used for object availability.	False
job.overwriteExistingObjects	When set to True, objects that already exist on the target database system are overwritten.	False
job.securityMgmtLevel	The level of security management enabled. Valid choices are Daemon and Job.	Job
job.useGroupUserIdPool	Defines a set of system names and credentials. When creating a job, this group user id pool can	None

Property	Description	Default Value
	be used for the source or target in place of directly specifying credentials in the job.	
<code>job.useSecurityMgmt</code>	When set to <code>True</code> , Data Mover commands require the admin username and password to be specified when running the command. For a complete list of commands affected by this parameter, see the <i>Teradata® Data Mover User Guide</i> .	<code>False</code>
<code>job.useSyncService</code>	Records any changes to the Data Mover repository tables (inserts/updates/deletes) in an audit log table. The value must be set to <code>True</code> to use the Sync service.	<code>False</code>
<code>job.useUserIdPool</code>	Uses a target user from the pool of users.	
<code>lighthouse.enable</code>	Enable or disable the Lighthouse data collection feature.	<code>True</code>
<code>lighthouse.hour</code>	The hour to begin the collection of the Data Mover matrixes. Default value 3 means 3 a.m.	3
<code>lighthouse.minute</code>	The minute to begin the collection of the Data Mover matrixes.	0
<code>map</code>	Represents the system-level map values for target systems with Teradata Database 16.10 or later. The map can be defined at the object, database, or job definition level. If the map is not defined, and objects are being copied to the target system as part of the job, then system-level maps are used for those objects in the target system. For more information, refer to About Teradata Database MAPS Architecture Feature Support in the <i>Teradata® Data Mover User Guide</i> .	<code>False</code>
<code>queryGridManagerEncryptedPassword</code>	Sets the QueryGrid Manager user-encrypted password. Cannot be combined with <code>queryGridManagerPassword</code> .	
<code>queryGridManagerPassword</code>	Sets the QueryGrid Manager user password. Cannot be combined with the <code>queryGridManagerEncryptedPassword</code> .	
<code>queryGridManagerUser</code>	Sets the QueryGrid Manager Manager user.	Support
<code>querygrid.wait.final.status</code>	When set to <code>True</code> , the system waits for QueryGrid Manager to return the final task status. Setting to <code>True</code> may impact system performance.	<code>False</code>
<code>repository.purge.definition.enabled</code>	Enables the automated purging of job definitions.	<code>False</code>
<code>repository.purge.enabled</code>	Enables/disables the repository clear feature. The default value <code>False</code> means the feature is disabled.	<code>False</code>
<code>repository.purge.history.unit</code>	The unit for job history data to be kept in the repository before purging occurs. The current supported values are Days, Weeks, Months, and Years.	Days

Property	Description	Default Value
<code>repository.purge.history.unitcount</code>	The number of units for job history data to be kept in the repository before purging occurs. This value is combined with the value for <code>repository.purge.history.unit</code> to determine the amount of time before purging occurs for old jobs (for example, 60 days, 3 years, or 10 months). The value of -1 disables the purging by time.	60
<code>repository.purge.hour</code>	The hour to start the daily repository purging. Default value 1 means 1 am.	1
<code>repository.purge.minute</code>	The minute to start the daily repository purging.	0
<code>repository.purge.percent</code>	The percentage of repository permSPACE that needs to be available to determine when to clear the repository. The default value 50 means clear the repository when more than 50% of the available permSPACE is in use. The value of -1 disables the purging by percentage.	50
<code>skip.FLML.tables</code>	Enables/disables a table from being Fast/Multi loaded, regardless of the utility being used. When enabled (true), the daemon marks the table as <code>completed_with_warnings</code> . When disabled (false) the daemon marks the table copy as <code>failed</code> .	False
<code>system.default.database.enabled</code>	Enables/disables the default target/staging databases at the system level. The default value False means disabled.	False
<code>target.system.load.slots</code>	Controls the total number of load slots that Data Mover can use at one time on target Teradata systems.	5
<code>tmsm.frequency.bytes</code>	Controls the frequency of messages sent to Teradata Ecosystem Manager when using byte-based utilities. Note: Providing a low value can hurt performance. Teradata recommends using the default value.	2147483647 BYTES
<code>tmsm.mode</code>	Controls how Data Mover directs Teradata Ecosystem Manager messages. Possible values are BOTH, ONLY_REAL_TMSM, ONLY_INTERNAL_TMSM, and NONE. When set to BOTH, messages are sent to the real Teradata Ecosystem Manager and written to the TDI event tables.	None

Configuring the Data Mover Agent

1. Edit the `agent.properties` file located in the `/etc/opt/teradata/datamover` directory and restart the Data Mover agent to implement the changes.

For properties that can be set dynamically, the changes take effect one minute after the updated `agent.properties` file is saved. There is no need to restart the agent service if you are only updating dynamic properties.

Installing and Configuring the Data Mover Agent on a Linux Teradata Server

The Data Mover Agent can be installed on a Linux Teradata server using the following procedure. You cannot use PUT to install or configure the Data Mover agent on a non-Teradata server, you must use `dminstallupgradeagent`.

1. Enter `./dminstallupgradeagent` at the command line to install the Data Mover agent and TTU packages.
2. Answer the prompts as needed, press **Enter** to accept the defaults where appropriate.
3. Enter `rpm -qa |grep DMAgent` to verify the installation.
4. Follow the steps in [Configuring ActiveMQ on Remote Agents](#) if this is the first time installing or upgrading a standalone agent using this script.
Once a standalone remote agent has been set up manually, the settings are preserved across upgrades.

```
*****
*
* WARNING: A new activemq encrypted password has been generated.
* Any remote Agents that are connecting to activemq on
* this server must be updated as follows:
*
* 1) Copy /etc/opt/teradata/tdactivemq/datamover.properties from
* this server to the same location on the remote Agent host.
*
* 2) Set the permissions of this file on the remote Agent as:
*     chmod 650 datamover.properties
*     chgrp activemq datamover.properties
*
* 3) Update Agent broker.password property in agent.properties
* with the value defined in agent.properties on this server.
*
* Note: Update /etc/opt/teradata/datamover/agent.properties
* if this file exists, otherwise update
* /opt/teradata/client/nn.mm/datamover/agent/agent.properties
* (where nn.mm = version of Datamover).
*
* 4) Restart the Agent:
*     /etc/init.d/dmagent stop
*     /etc/init.d/dmagent start
*
```

The agent.properties File

Property	Description	Default Value
<code>agent.id=id</code>	Unique identifier for this agent.	Agent1
<code>cluster.enabled=setting for cluster</code>	When set to True , establishes a connection to a standby Java Message Service (JMS) broker in case the active JMS broker fails.	False
<code>broker.port=port number</code>	The port number of the machine where the Java Message Service (JMS) Message Broker is listening.	61616
<code>broker.url=url</code>	The hostname or IP address of the machine running the Java Message Service (JMS) Message Broker.	localhost
<code>rootLogger.level</code>	The six levels of logging, TRACE DEBUG INFO WARN ERROR FATAL. From trace level to application error. LOG_LEVEL can be updated dynamically, but not logfile. ¹	INFO
<code>appender.rolling.type</code>	The six levels of logging, TRACE DEBUG INFO WARN ERROR FATAL. From trace level to application error. LOG_LEVEL can be updated dynamically.	RollingFile
<code>appender.rolling.name</code>	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
<code>appender.rolling.layout.type</code>	Do not edit. This is an internal setting for logging infrastructure.	PatternLayout
<code>appender.rolling.layout.pattern</code>	Do not edit. This is an internal setting for logging infrastructure. <ul style="list-style-type: none"> d = date t = thread p = log level c = class name m = message n = platform dependent line separator 	%d [%t] %-5p %c{3}(%L) - %m%n
<code>appender.rolling.policies.type</code>	Do not edit. This is an internal setting for logging infrastructure.	Polices
<code>appender.rolling.policies.size.type</code>	Do not edit. This is an internal setting for logging infrastructure.	SizeBasedTriggeringPolicy
<code>appender.rolling.strategy.type</code>	Do not edit. This is an internal setting for logging infrastructure.	DefaultRolloverStrategy
<code>logger.rolling.name</code>	Do not edit. This is an internal setting for logging infrastructure.	com.teradata.datamovement.agent
<code>logger.rolling.appenderRef.rolling.ref</code>	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
<code>appender.rolling.fileName</code>	The relative or absolute path of the log file. If changing the location, specify the absolute path of the file.	/var/opt/teradata/datamover/logs/dmAgent.log

Property	Description	Default Value
	For Windows, specify back slash instead of forward slash, for example, C:\Program File\Teradata\Log\dmAgent.log. Dynamic property. ¹	
appender.rolling.filePattern	Specifies logfiles pattern. Example: dmAgent.log.%i saves the files as dmAgent.log.1, dmAgent.log.2, dmAgent.log.3, and so on.	/var/opt/teradata/datamover/logs/dmAgent.log.%i
appender.rolling.policies.size.size	The maximum size of the logging file before being rolled over to backup files. Dynamic property. ¹	10MB
appender.rolling.strategy.max	The number of backup logging files that are created. After the maximum number of files have been created, the oldest file is deleted. Dynamic property. ¹ Example: If maximum backups = 3, three backup logs are created: <ul style="list-style-type: none"> • dmAgent.log.1 • dmAgent.log.2 • dmAgent.log.3 If the current dmAgent.log size exceeds 20MB, it rolls to become the new dmAgent.log.1 and a new dmAgent.log is created. The previous dmAgent.log.2 becomes the new dmAgent.log.3. The previous dmAgent.log.3 is deleted.	3
agent.maxConcurrentTasks= <i>maximum number of tasks</i>	The maximum number of tasks allowed to run on this agent at the same time. Note that tasks are distributed to agents using a round robin method. Task size is not currently considered, so load may not be balanced if one agent is randomly assigned larger tasks than another.	5
tvi.useLogger= <i>setting for TVI messages</i>	The TVI logger can be set to true or false. If set to true, fatal error messages are sent to TVI. Dynamic property. ¹	True

¹For properties that can be set dynamically, the changes take effect one minute after the updated daemon.properties file is saved. There is no need to restart the daemon service if you are only updating dynamic properties. For example:

- If you changed the value of rootLogger.level from the default of INFO, logfile to DEBUG, logfile, any debug messages generated would start appearing in the log file one minute after saving the updated properties file.
- If you changed the value of agent.maxConcurrentTasks from the default value of 5 to a new value of 6, the new value of 6 would take effect one minute after saving the updated agent.properties file.

Configuring the Data Mover Command-Line Interface

Configuring the Data Mover Command-Line Interface on a Linux Teradata Server

The Data Mover Command-Line Interface is installed for Linux Teradata servers with PUT. Configure the command line properties to customize these settings.

1. Edit the `commandline.properties` file located in the `/etc/opt/teradata/datamover` directory to customize the command line properties.

Installing and Configuring the Data Mover Command-Line Interface on Non-Teradata Servers

The Data Mover Command-Line Interface must be installed for Linux on non-Teradata servers, Windows, Solaris Sparc, Ubuntu, and IBM AIX systems using the following procedure. You cannot use PUT to install the Command-Line Interface on those systems.

Steps 1 through 4 do not apply to installation on Windows systems.

1. Add the following lines of code to the end of the `/etc/profile` file to update the `JAVA_HOME` and `PATH` environment variables for all users:

```
export JAVA_HOME={full path of java installation location}
export PATH=$JAVA_HOME/bin:$PATH
```
2. Run the command:

```
source /etc/profile
```
3. Verify that the output shows JRE.1.8:

```
java -version
```
4. Open the `.profile` file of the root user and verify that the values for the `JAVA_HOME` and `PATH` environment variables are the same as those defined in `/etc/profile`.
 If the values are different, the `java -version` command will not produce the correct output during install time, and the installation will fail.
5. Install the appropriate `DMCmdline` software package for your system as follows:

Operating System	Actions
Linux (for non-Teradata servers)	<ol style="list-style-type: none"> a. At the command line, type <code>export DM_INTERACTIVE_INSTALL=1</code> to set the environment variable for interactive install. b. At the command line, type the following: <pre>gunzip DMCmdline__linux_i386.17.12.xx.xx.tar.gz tar xvf DMCmdline__linux_i386.17.12.xx.xx.tar cd DMCmdline.17.12* rpm -Uvh DMCmdline__linux_noarch.17.12`.xx.xx-1.rpm</pre>

Operating System	Actions
	<ul style="list-style-type: none"> c. Answer the prompts as needed and press Enter to accept the defaults where appropriate. d. Type <code>rpm -qa grep DMCmdline</code> to verify the installation.
Windows	<ul style="list-style-type: none"> a. Copy the Data Mover directory on the media to a folder on the hard drive. b. Go to DataMover/Windows and unzip <code>tdm-windows__windows_i386.17.12.xx.xx.zip</code>. c. Go to the DISK1 directory and run <code>setup.exe</code>. d. Answer the prompts as needed and press Next to accept defaults where appropriate. e. Select Install when finished. f. Go to Start > Control Panel > Add or Remove Programs to verify installation.
Solaris Sparc	<ul style="list-style-type: none"> a. At the command line, type the following to install: <code>gunzip tdm-solaris__solaris_sparc.17.12.xx.xx.tar.gz</code> <code>tar xvf tdm-solaris__solaris_sparc.17.12.xx.xx.tar</code> <code>pkgadd -d 'pwd' DMCmdline</code> b. Answer the prompts as needed and press Enter to accept defaults where appropriate. c. Type <code>pkginfo -l DMCmdline</code> to verify the installation.
IBM AIX	<ul style="list-style-type: none"> a. At the command line, type the following to install: <code>gunzip tdm-aix__aix_power.17.12.xx.xx.tar.gz</code> <code>tar xvf tdm-aix__aix_power.17.12.xx.xx.tar</code> <code>installp -acF -d ./DMCmdline DMCmdline</code> b. Answer the prompts as needed and press Enter to accept defaults where appropriate. c. Type <code>lsllpp -l "DM*"</code> to verify the installation.
Ubuntu	<ul style="list-style-type: none"> a. At the command line, type <code>export DM_INTERACTIVE_INSTALL=1</code> to set the environment variable for interactive install. b. At the command line, type the following: <code>tar xzvf tdm-ubuntu__ubuntu.17.12.xx.xx.tar.gz</code> <code>cd DMCmdline.17.12.xx.xx</code> <code>dpkg -i DMCmdline__ubuntu_all.17.12.xx.xx-1.deb</code> Note: In Ubuntu, <code>-i</code> is used for both install and upgrade. c. Answer the prompts as needed and press Enter to accept the defaults where appropriate. d. Type <code>dpkg -l grep dmcmdline</code> to verify the installation.

6. If the REST server needs to be changed, edit the `commandline.properties` file located in the `TDM_install_directory\CommandLine\commandline.properties` directory after installation.

7. Specify the Data Mover REST server URL for communicating with the daemon as in the following example:

```
dm.rest.endpoint=https://dm_host:1443/datamover
```

Make sure the `host:port` value used for `dm.rest.endpoint` is on the `accept.host.list` in `tdmrest.properties`.

The commandline.properties File

Property	Description	Default Value
<code>dm.rest.endpoint</code>	The Data Mover REST server URL. When automatic failover support is configured, use a comma separated list to add the standby REST server URL. Example: <code>dm.rest.endpoint=https://activeServer:1443/datamover,https://standbyServer:1443/datamover</code> Make sure the <code>host:port</code> value used for <code>dm.rest.endpoint</code> is on the <code>accept.host.list</code> in <code>tdmrest.properties</code> .	<code>https://localhost:1443/datamover</code>
<code>rootLogger.level</code>	The six levels of logging, TRACE DEBUG INFO WARN ERROR FATAL. From trace level to application error. LOG_LEVEL can be updated dynamically, but not logfile. ¹	INFO
<code>appender.rolling.type</code>	The six levels of logging, TRACE DEBUG INFO WARN ERROR FATAL. From trace level to application error. LOG_LEVEL can be updated dynamically.	RollingFile
<code>appender.rolling.name</code>	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
<code>appender.rolling.layout.type</code>	Do not edit. This is an internal setting for logging infrastructure.	PatternLayout
<code>appender.rolling.layout.pattern</code>	Do not edit. This is an internal setting for logging infrastructure. <ul style="list-style-type: none"> d = date t = thread p = log level c = class name m = message n = platform dependent line separator 	<code>%d [%t] %-5p %c{3}(%L) - %m%n</code>
<code>appender.rolling.policies.type</code>	Do not edit. This is an internal setting for logging infrastructure.	<code>com.teradata.datamovement.commandline</code>
<code>appender.rolling.policies.size.type</code>	Do not edit. This is an internal setting for logging infrastructure.	SizeBasedTriggeringPolicy
<code>appender.rolling.strategy.type</code>	Do not edit. This is an internal setting for logging infrastructure.	DefaultRolloverStrategy
<code>logger.rolling.name</code>	Do not edit. This is an internal setting for logging infrastructure.	
<code>logger.rolling.appenderRef.rolling.ref</code>	Do not edit. This is an internal setting for logging infrastructure.	RollingFile

Property	Description	Default Value
appender.rolling.fileName	The relative or absolute path of the log file. If changing the location, specify the absolute path of the file. For Windows, specify back slash instead of forward slash, for example, C:\Program File\Teradata\Log\dmCommandLine.log. Dynamic property. ¹	dmCommandLine.log
appender.rolling.filePattern	Specifies logfiles pattern. Example: dmCommandLine.log.%i saves the files as dmCommandLine.log.1, dmCommandLine.log.2, dmCommandLine.log.3, and so on.	dmCommandLine.log.%i
appender.rolling.policies.size.size	The maximum size of the logging file before being rolled over to backup files. Dynamic property. ¹	20MB
appender.rolling.strategy.max	The number of backup logging files that are created. After the maximum number of files have been created, the oldest file is deleted. Dynamic property. ¹ Example: If maximum backups = 3, three backup logs are created: <ul style="list-style-type: none"> • dmCommandLine.log.1 • dmCommandLine.log.2 • dmCommandLine.log.3 If the current dmCommandLine.log size exceeds 20MB, it rolls to become the new dmCommandLine.log.1 and a new dmCommandLine.log is created. The previous dmCommandLine.log.2 becomes the new dmCommandLine.log.3. The previous dmCommandLine.log.3 is deleted.	3

Configuring the Data Mover REST Service

During the installation of Data Mover, the Data Mover REST component is installed and started automatically. You need to configure the `tdmrest.properties` file for your environment and restart the service.

Data Mover REST and DSA REST, which is bundled with Data Mover, have HTTPS enabled by default allowing you to make REST calls over HTTPS. If you try to use HTTP for a Data Mover REST job after a fresh installation or upgrade of Data Mover, the job is redirected to HTTPS.

1. In the directory `/etc/opt/teradata/datamover`, locate `tdmrest.properties`.
2. Configure the following properties:

Property	Description	Default Value
accept.host.list	If requests to the Data Mover REST API need to use different values compared to the default for the host header or X-Forwarded-Host, specify those values here. Leave localhost:1443 as the first value unless you do not want to allow calls to Data Mover REST API to use localhost:1443 as in the following example: accept.host.list=localhost:1443, host1:port, host2	server name:1443 or server IP:1443

Property	Description	Default Value
appender.rolling.fileName	The relative or absolute path of the log file.	/var/opt/teradata/datamover/logs/dmRest.log
appender.rolling.filePattern	Specifies logfiles pattern. Example: dmRest.log.%i saves the files as dmRest.log.1, dmRest.log.2, dmRest.log.3, and so on.	/var/opt/teradata/datamover/logs/dmRest.log.%i
appender.rolling.layout.type	Do not edit. This is an internal setting for logging infrastructure.	PatternLayout
appender.rolling.layout.pattern	Do not edit. This is an internal setting for logging infrastructure.	%d [%t] %-5p %c{3}(%L) - %m%ns
appender.rolling.name	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
appender.rolling.policies.size.type	Do not edit. This is an internal setting for logging infrastructure.	SizeBasedTriggeringPolicy
appender.rolling.policies.size.size	The maximum size of the logging file before being rolled over to backup files.	20MB
appender.rolling.policies.type	Do not edit. This is an internal setting for logging infrastructure.	Policies
appender.rolling.strategy.max	The number of backup logging files that are created. After the maximum number of files have been created, the oldest file is deleted. Example: If maximum backups = 3, three backup logs are created: <ul style="list-style-type: none"> dmRest.log.1 dmRest.log.2 dmRest.log.3 If the current dmRest.log size exceeds 20MB, it rolls to become the new dmRest.log.1 and a new dmRest.log is created. The previous dmRest.log.2 becomes the new dmRest.log.3. The previous dmRest.log.3 is deleted.	5
appender.rolling.strategy.type	Do not edit. This is an internal setting for logging infrastructure.	DefaultRolloverStrategy
appender.rolling.type	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
broker.url=url	Hostname or IP address of the machine running the Java Message Service (JMS) message broker.	localhost
broker.port=port	Port number of the machine where the JMS message broker is listening.	61616
cluster.enabled=setting	When set to true, a connection is established to a standby JMS broker for the cluster if active JMS broker fails.	False
logger.rolling.appenderRef.rolling.ref	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
logger.rolling.name	Do not edit. This is an internal setting for logging infrastructure.	com.teradata.datamovement.rest

Property	Description	Default Value
response.timeout	If progress of the job is not reported within this period (in seconds), the job is aborted.	30 sec
rootLogger.level	The six levels of logging, TRACE DEBUG INFO WARN ERROR FATAL. From trace level to application error.	INFO

- Restart the Data Mover REST service:
`/etc/init.d/tdmrest start`

Configuring the Cloud Staging Copy REST Service

During the installation of Data Mover, the Cloud Staging Copy Service (CS2) REST component is installed and started automatically. You need to configure the `cs2.properties` file for your environment and restart the service.

- In the directory `/etc/opt/teradata/datamover`, locate `cs2.properties`.
- Configure the following properties:

Property	Description	Default Value
dsa.rest.endpoint	The DSA REST URL used by CS2 for the DSA utility. You can override the default value by adding the same named property in the <code>cs2.properties</code> file.	<code>https://localhost:9090/dsa</code>
rootLogger.level	The six levels of logging from trace level to application error. <ul style="list-style-type: none"> TRACE DEBUG INFO WARN ERROR FATAL 	INFO
rootLogger.appendRef.rolling.ref	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
appender.rolling.type	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
appender.rolling.name	Do not edit. This is an internal setting for logging infrastructure.	RollingFile
appender.rolling.layout.type	Do not edit. This is an internal setting for logging infrastructure.	PatternLayout
appender.rolling.layout.pattern	Do not edit. This is an internal setting for logging infrastructure.	<code>%d [%t] %-5p %c{3}(%L) - %m%n</code>
appender.rolling.policies.type	Do not edit. This is an internal setting for logging infrastructure.	Policies
appender.rolling.policies.size.type	Do not edit. This is an internal setting for logging infrastructure.	SizeBasedTriggeringPolicy

Property	Description	Default Value
appender.rolling.strategy.type	Do not edit. This is an internal setting for logging infrastructure.	DefaultRolloverStrategy
appender.rolling.fileName	The relative or absolute path of the log file.	/var/opt/teradata/datamover/logs/dmCS2.log
appender.rolling.filePattern	Specifies logfiles pattern. Example: dmCS2.log.%i saves the files as dmCS2.log.1, dmCS2.log.2, dmCS2.log.3, and so on.	/var/opt/teradata/datamover/logs/dmCS2.log.%i
appender.rolling.policies.size.size	The maximum size of the logging file before being rolled over to backup files.	10 MB
appender.rolling.strategy.max	The number of backup logging files that are created. After the maximum number of files have been created, the oldest file is deleted. Example: If maximum backups = 3, three backup logs are created: <ul style="list-style-type: none"> • dmCS2.log.1 • dmCS2.log.2 • dmCS2.log.3 If the current dmCS2.log file size exceeds 10MB, the file rolls to become the new dmCS2.log.1, and: <ul style="list-style-type: none"> • The previous dmCS2.log.1 becomes the new dmCS2.log.2. • The previous dmCS2.log.2 becomes the new dmCS2.log.3. • The previous dmCS2.log.3 is deleted. 	3
rest.connection.timeout	Defines the timeout in milliseconds for a connection to be established with DSA REST.	30000
rest.read.timeout	Defines the socket timeout in milliseconds when sending requests to DSA REST.	120000
swagger.ui.enabled	Enable/Disable Swagger user interface	False

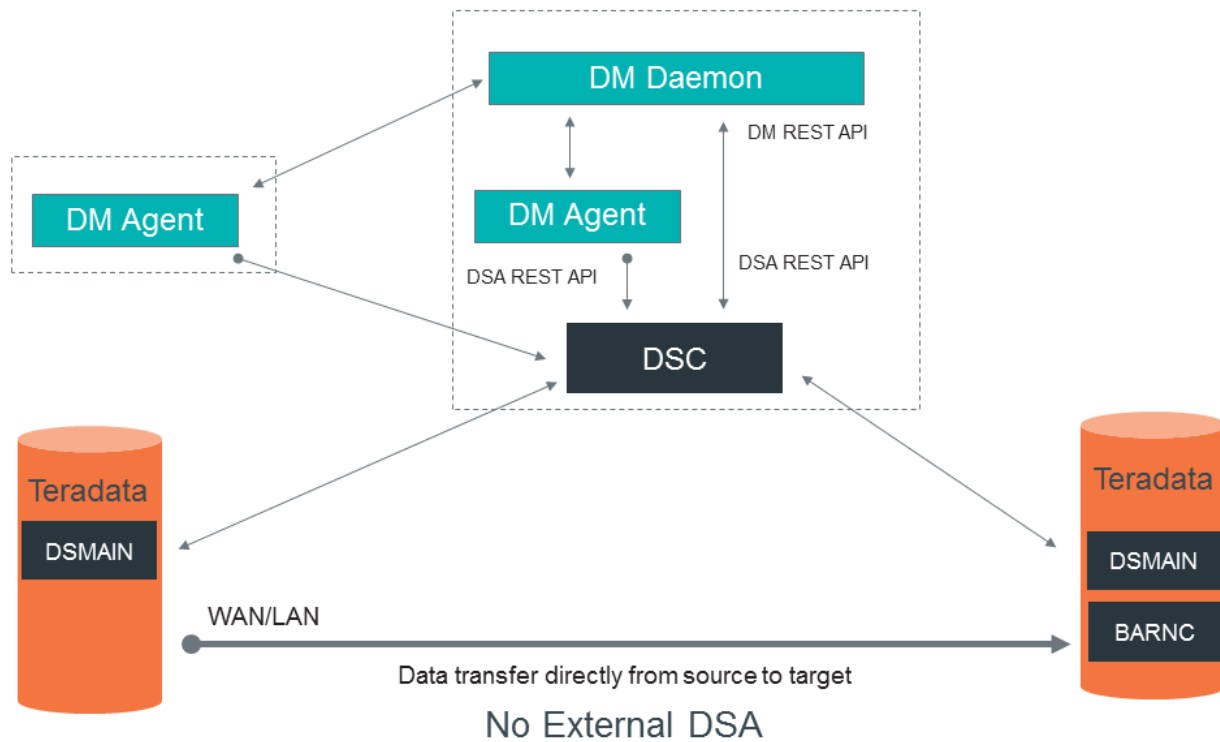
3. Restart the Cloud Staging Copy Service: `/etc/init.d/dmcs2 restart`

DSA Configurations

There are three main supported configurations for running Data Mover DSA jobs:

Configuration 1: DSC Running on the Data Mover Server

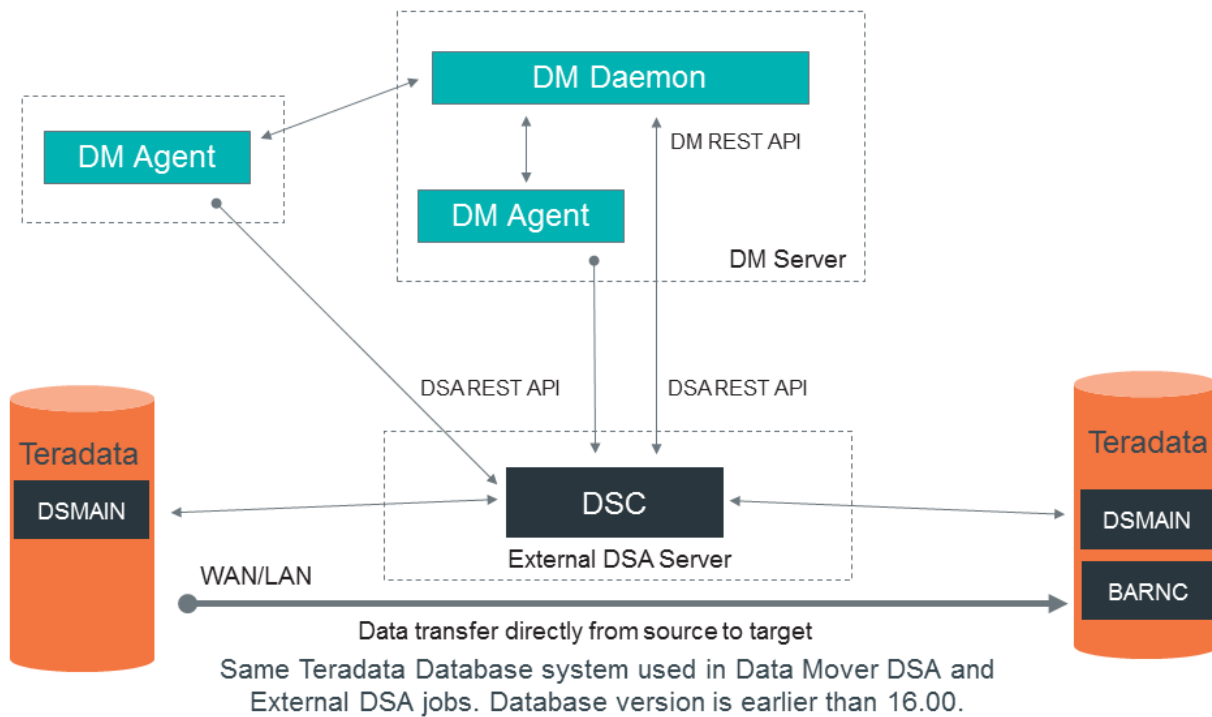
Configuration 1 is the default setup when you do not have an existing DSA environment. The bundled DSC that comes with Data Mover is enabled for use.



See [Configuring DSC to Run on the Data Mover Server](#) for configuration instructions.

Configuration 2: Data Mover Using an External DSC

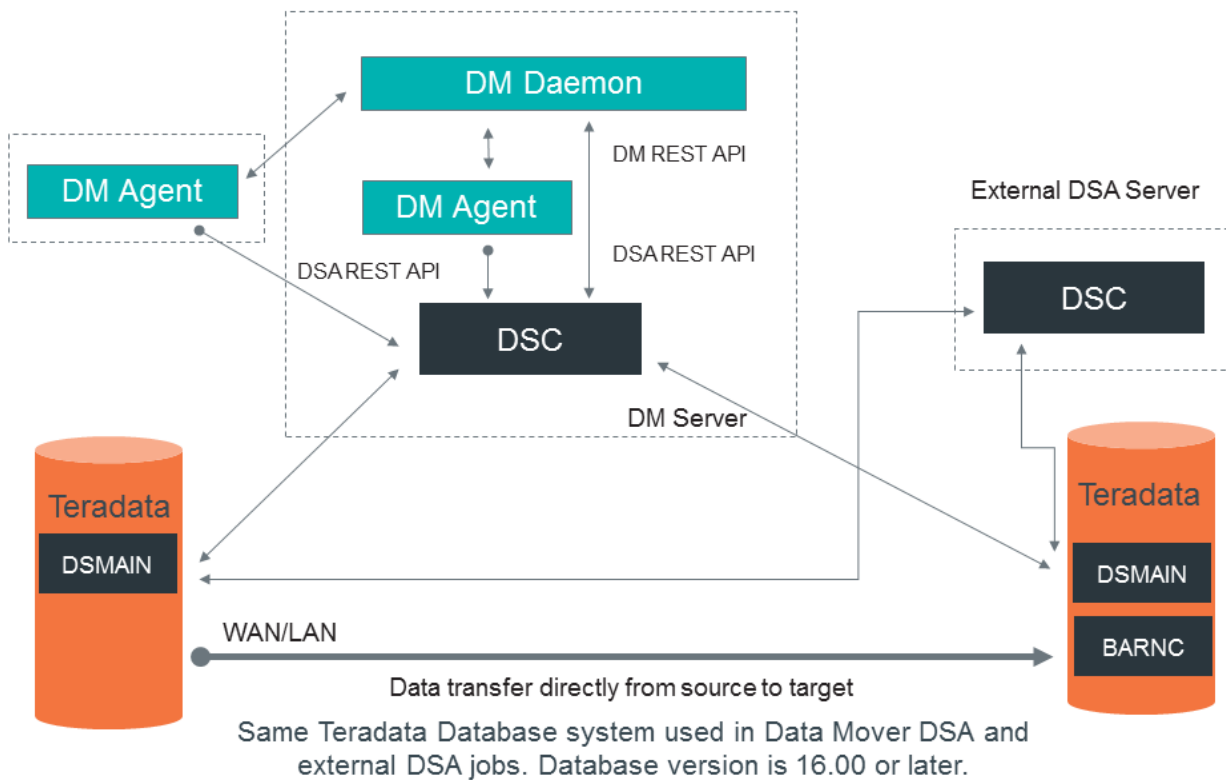
Use configuration 2 when you have an external DSC and one or more of your Teradata Data Mover source or target systems is running a version earlier than 16.00.



See [Configuring Data Mover to Use an External DSC](#) for configuration instructions.

Configuration 3: DSC Running on the Data Mover Server and Externally

Use configuration 3 when you have an external DSC and all your Teradata Data Mover source and target systems are version 16.00 and later. In this case, both the external DSC and the Data Mover bundled DSC can coexist.



See [Configuring DSC to Run on the Data Mover Server and Externally](#) for configuration instructions.

Configuring DSC to Run on the Data Mover Server

Refer to the following configuration steps when running DSA on the Data Mover server. This only applies to systems using Data Mover 16.20 or later.

1. [Register the source and target systems with DSC and restart DSMMAIN.](#)
2. [Install and configure BAR NC on the source or target TPA nodes.](#)
3. [Configure Network Fabric or Logical Netmask.](#)
4. [Enabling TLS 1.2 Data Path Encryption for DSC on the Data Mover Server](#)

Configuring Data Mover to Use an External DSC

Refer to the following configuration steps when running DSA on an external server.

1. Configure the Data Mover daemon to use an external DSC by changing the `dameon.properties` file.
 - a. Specify that DSC is not colocated with the Data Mover daemon by adding the `is.dsc.colocate.dm=false` property.
 - b. Specify the location of the DSA REST server by adding the `dsa.rest.endpoint=https://external_dsc_host:9090/dsa` URL.
If the external DSA REST server is configured with HTTP, use the `dsa.rest.endpoint=http://external_dsc_host:9090/dsa` URL instead. Do not change

the default `dm.rest.endpoint=https://localhost:1443/datamover` URL unless instructed to do so by Teradata Customer Support.

If `dm.rest.endpoint` needs to be modified, verify that the hostname used is listed as one of the accepted hosts for the `accept.host.list` property in `tdmrest.properties`.

- c. Restart both the Data Mover daemon and agents after making the DSA property changes.
2. Copy the `build_dsainputs` and `dsa_configsys` tools to the external DSC server.
3. For new source or target systems:
 - a. [Register the source and target systems with DSC and restart DSMMAIN.](#)
 - b. [Install and configure BAR NC on the source or target TPA nodes.](#)
 - c. [Configure the logical netmask.](#)
4. If you need to configure the [Cloud Staging Copy Service](#) for using external DSC, update the `cs2.properties` file as following:
 - a. Specify the location of the DSA REST server by adding the URL: `dsa.rest.endpoint=https://external_dsc_host:9090/dsa`.
 - b. Restart the Cloud Staging Copy Service:

```
/etc/init.d/dmcs2 restart
```

Configuring DSC to Run on the Data Mover Server and Externally

Refer to the following configuration steps when running DSA on both the Data Mover server and on an external server. To run DSA on the Data Mover server, you must have Data Mover 16.20 or later installed.

1. For source and target TPA nodes with existing BAR NC ClientHandler software, edit the `/etc/opt/teradata/dsa/clienthandler.properties` and append the Data Mover host to `broker.list` as in the following example:

```
broker.list=153.64.24.162:61616,153.64.24.164:61616
```

2. [Register all source and target systems with DSC and restart DSMMAIN.](#)
3. [Install and configure BAR NC on new source or target TPA nodes.](#)
4. [Configuring the Logical Netmask for ClientHandler.](#)
5. [Enabling TLS 1.2 Data Path Encryption for DSC on the Data Mover Server](#)

Backing Up the DSA Repository

To back up the repository, see "Protecting the DSC Repository" in *Teradata® DSA User Guide*, B035-3150.

Cloud Staging Copy Service with Data Mover

The Cloud Staging Copy Service (CS2) is a REST service packaged with Data Mover. The Cloud Staging Copy Service provides a method of copying data from a source system to a target system through cloud storage. The name of this cloud storage is Cloud Staging Area.

Note:

Currently the Data Mover supports only AWS S3 cloud storage.

The Cloud Staging Copy Service copies data in the following three steps:

1. Takes data backup from the source system to the Cloud Staging Area.
2. Copies data from the Cloud Staging Area to the target system.
3. Cleans up the backup data from the Cloud Staging Area after the copy task completes.

To take back up to and restore from cloud storage, the Cloud Staging Copy Service uses DSA. This service interacts with DSC to handle the following:

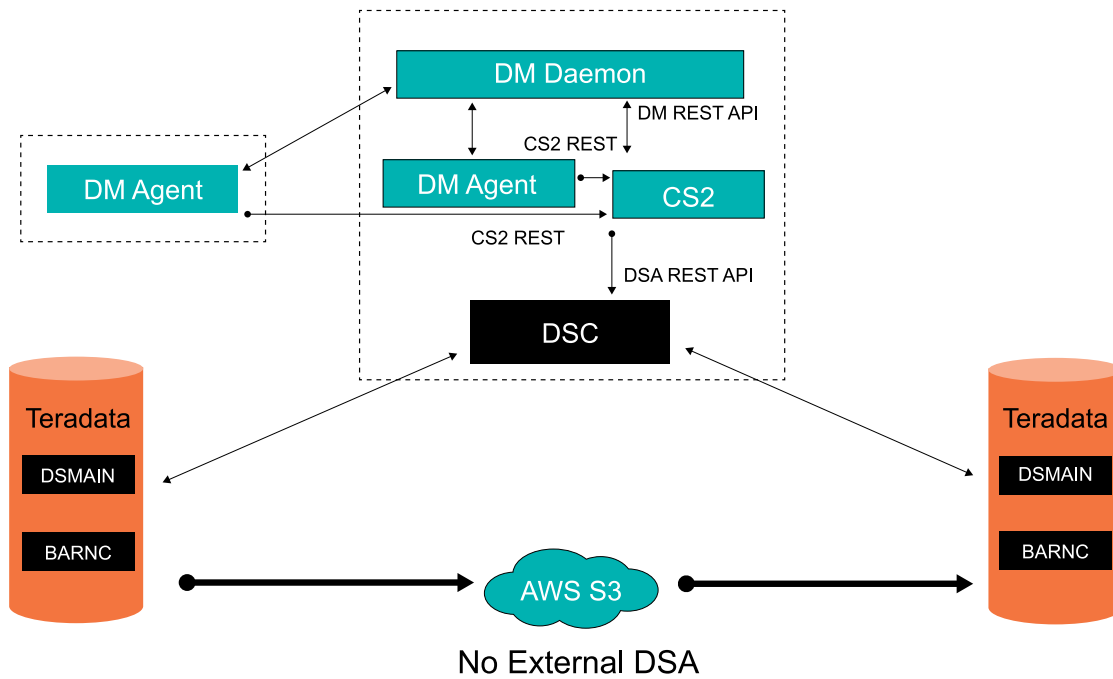
- Configuring DSC to copy data from the source to the Cloud Staging Area, and from the Cloud Staging Area to the target.
- Creating and executing a DSA backup job to back up data from the source to the Cloud Staging Area.
- Creating and executing DSA restore job to restore data from the Cloud Staging Area to the target.
- Calling DSA to cleanup backup data from the Cloud Staging Area after the DSA restore job completes.

The Cloud Staging Copy Service runs as a service embedded on the Data Mover host. The service provides a REST API that uses port 3443. You do not have to interact directly with the Cloud Staging Copy Service. You interact with the Data Mover interfaces. The Data Mover Daemon and Agent handle the interaction with the Cloud Staging Copy Service. Port 3443 only needs to be open to incoming traffic on the Data Mover host if there is an external Data Mover Agent. Otherwise, all traffic to port 3443 is local.

There are three main supported configurations for running Data Mover DSA jobs with the Cloud Staging Copy Service:

Configuration 1: DSC Running with Cloud Staging Copy Service on the Data Mover Server

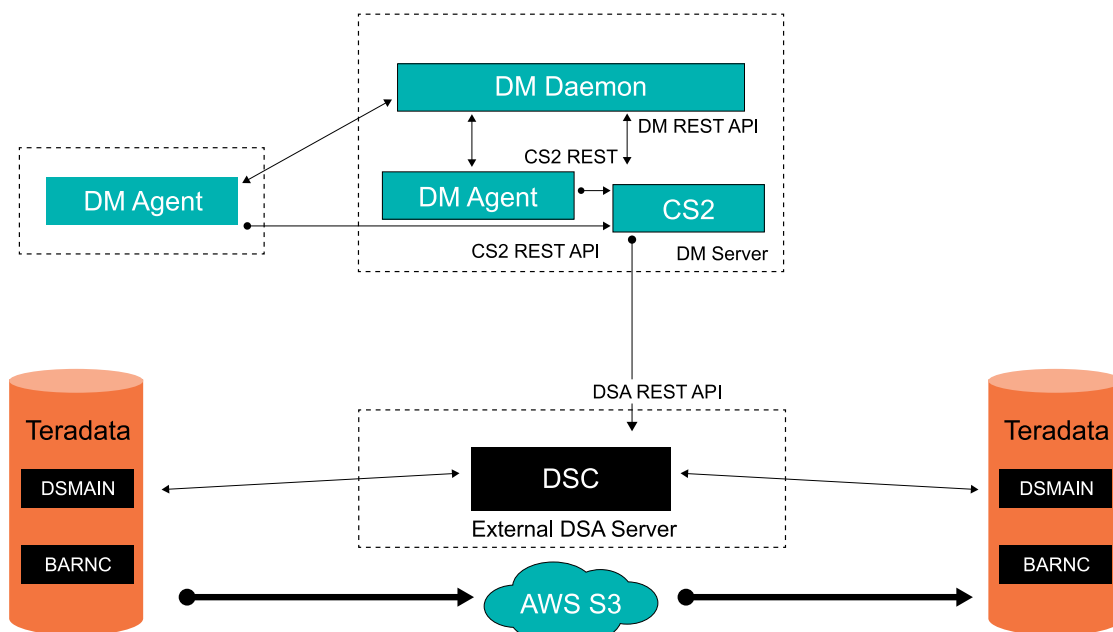
Configuration 1 is the default setup when you do not have an existing DSA environment. The bundled DSC and the Cloud Staging Copy Service come with Data Mover are enabled for use.



See [Configuring DSC to Run on the Data Mover Server](#) for configuration instructions.

Configuration 2: Data Mover Using Cloud Staging Copy Service and an External DSC

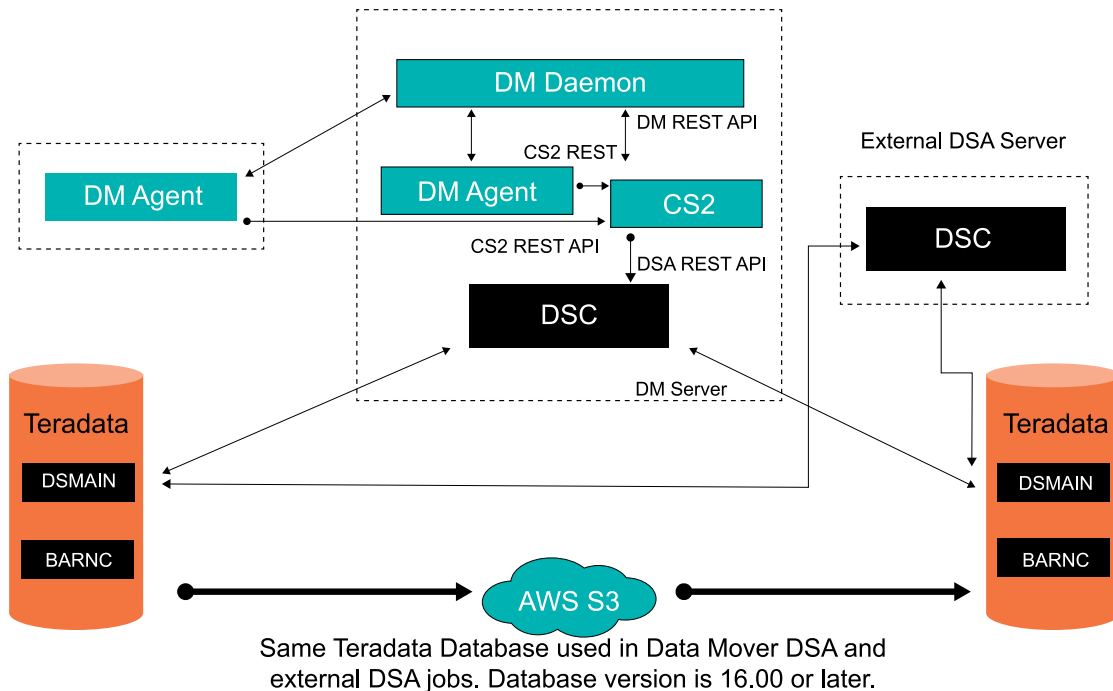
Use configuration 2 when you have an external DSC, one or more of your Data Mover source or target systems is running a version earlier than 16.00, and the Cloud Staging Copy Service is enabled for use on the host.



See [Configuring Data Mover to Use an External DSC](#) for configuration instructions.

Configuration 3: DSC Running with Cloud Staging Copy Service on the Data Mover Server and Externally

Use configuration 3 when you have an external DSC, all your Data Mover source and target systems are version 16.00 and later, and the Cloud Staging Copy Service is enabled for use on the host. Here, both the external DSC and the Data Mover bundled DSC can coexist.



See [Configuring DSC to Run on the Data Mover Server and Externally](#) for configuration instructions.

Enabling Cloud Staging Copy Service for Data Mover Jobs

To enable the Cloud Staging Copy Service for Data Mover Jobs:

1. [Register the source and target systems with DSC and restart DSMAIN.](#)
2. [Install and configure BAR NC on the source or target TPA nodes.](#)
3. [Install AXMS3 Module on the source and target TPA nodes.](#)
4. [Set up Cloud Staging Area.](#)

High Availability Overview

A high availability configuration is the base configuration for a Data Mover system. If a primary (designated-active) component of a system becomes unavailable, a high availability configuration makes sure that the system continues to function with components running on the secondary (designated-standby) system.

A high availability configuration depends on a monitoring service, which monitors the actively running components and services through SSH connections. If any of the designated-active services stop running, the failover monitor first attempts to rescue the service. If rescue attempts fail, a failover sequence begins the process of allowing the designated-standby component to take over for the designated-active component.

The following monitoring service requirements apply in a high availability environment:

- The daemon, agent, Data Mover REST, ActiveMQ, and sync monitor service on the active and standby components are run using user `dmuser`
- The monitoring service cannot be used to monitor Data Mover components if a user other than `dmuser` has been set up to run these services.
- The bundled DSA service must be run with the user `dscuser`.

Configuring Automatic Failover

Data Mover provides automatic failover support when multiple Data Mover servers are configured in a dual environment. Automatic failover configuration must meet the following requirements:

- Two additional monitoring servers available to monitor the active and standby components. It is highly recommended that you use a Viewpoint multi-purpose server for this configuration.
- Each monitoring server must be local to the site and ideally be attached to the same network as the components being monitored to avoid an automatic failover caused by network partitions.
- The DMFailover package must be installed on all servers, including the active and standby daemon, monitoring, and agent servers that are part of the cluster.

If additional monitoring servers are not available, you can enable repository synchronization by using the Data Mover synchronization service. The sync service alone does not support automatic failover and requires manual intervention to enable failover from active to standby components. If configuring the sync service without configuring automatic failover, see [Configuring the Synchronization Service without Automatic Failover](#).

Note:

When using the synchronization service with failover, use the public IP address or hostname of the system, not the `localhost` hostname or the `127.0.0.1` IP address.

The following files, located in the `/opt/teradata/client/nn.nn/datamover/failover/` folder, are required when using the monitoring service. Where *nn.nn* in the filename refers to the Data Mover version.

File	Description
<code>/etc/opt/teradata/datamover/failover.properties</code>	Specifies the active and standby components to be monitored.
<code>/opt/teradata/client/<i>nn.nn</i>/datamover/failover/dmcluster</code>	Script for setting up SSH log on, configuring the servers in active and standby modes, starting and stopping the monitoring service, and checking the status of the active and standby components.

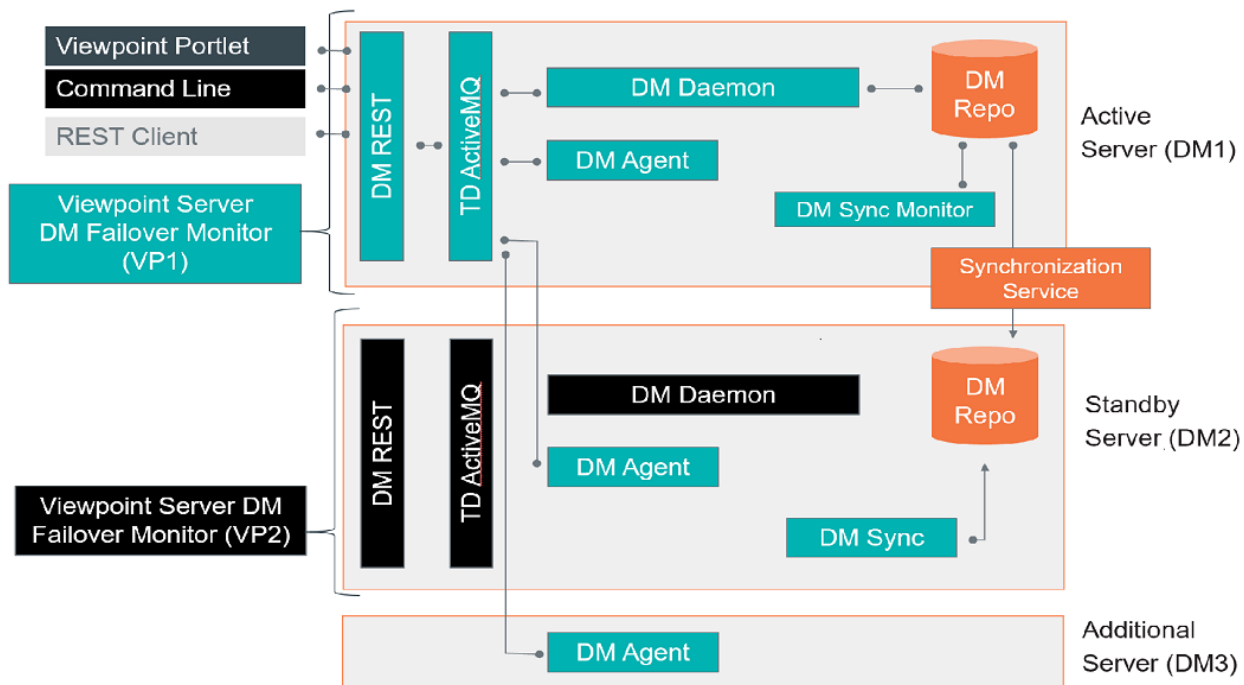
File	Description
/opt/teradata/client/nn.nn/datamover/failover/DMFailover.jar	Executable binary file used for automatic failover.
/etc/opt/teradata/datamover/monitor.properties	Specifies if Server Management alerts need to be sent if the monitoring service detects a failure. This file is used only on the monitoring server.

The following tasks must be performed to configure automatic failover:

1. [Verifying Data Mover Package Installation](#)
2. [Setting Up Host Files or DNS](#)
3. [Verifying Required Ports Open](#)
4. [Defining Unique Data Mover Agent Names](#)
5. [Configuring Cluster Settings](#)
6. [Configuring the Cluster and Starting the Monitoring Service](#)
7. [Checking the Status of Active and Standby Components](#)
8. [Verifying Failover Configuration](#)

High Availability Configuration Scenario

To configure high availability successfully, each step must be performed in order as they appear in this section. After each step, a use-case example to demonstrate the configuration process is provided. The following figure and table describe the terms used during the failover configuration process.



Example Server Name	Server Type	Description
DM1	Designated-active	The system primarily used as the active system before failover occurred. Typically, the server where the primary Data Mover ActiveMQ daemon, agent, and REST reside.
DM2	Designated-standby	The standby system assigned to take responsibility as the active system if the designated-active stops working. If the DM1 server fails, this is the server that becomes the next active. In the meantime, one Data Mover agent runs here.
DM3	Additional agent	Server where one Data Mover agent runs. Not a standby server.
VP1	Designated-active monitor	Server where the primary Data Mover monitor runs. This monitor continuously checks that DM1 is running properly and initiates failover to DM2 if necessary.
VP2	Designated-standby monitor	Server where the secondary Data Mover monitor runs. This monitor only becomes active if failover occurs. If failover occurs, this monitor continuously checks that DM2 is running properly. There is no failover if failure occurs. If Data Mover is configured to log to server management, TVI failure alerts are generated.

Verifying Data Mover Package Installation

1. Verify the designated-active, designated-standby, and standby-only servers have the following required packages installed:
 DMAgent-XX.XX.XX.XX-1
 DMCmdline-XX.XX.XX.XX-1
 DMDaemon-XX.XX.XX.XX-1
 DMFailover-XX.XX.XX.XX-1
 tdmrest-XX.XX.XX.XX-1
 DMSync-XX.XX.XX.XX-1

 Servers must be one of the following server types:
 - Data Mover Teradata Multi-Purpose Server
 - Consolidated Teradata Multi-Purpose Server with Data Mover
2. Verify the additional agent server has the DMAgent-XX.XX.XX.XX-1 package installed.
 Server must be one of the following server types:
 - Data Mover Teradata Multi-Purpose Server
 - Consolidated Teradata Multi-Purpose Server with Data Mover
 - Cloud-based instance of Data Mover (for example, Teradata Managed Cloud or AWS)
3. Verify the designated-active monitor and the designated-standby monitor have the DMFailover-XX.XX.XX.XX-1 package installed.
 Viewpoint Teradata Multi-Purpose Server is recommended.

Example: Verifying Data Mover Package Installation

1. On DM1 and DM2, verify that the following packages are installed:
 - DMAgent-XX.XX.XX.XX-1
 - DMCmdline-XX.XX.XX.XX-1
 - DMDaemon-XX.XX.XX.XX-1
 - DMFailover-XX.XX.XX.XX-1
 - DMSync-XX.XX.XX.XX-1
 - tdmrest-XX.XX.XX.XX-1
2. On DM3, verify that the DMAgent-XX.XX.XX.XX-1 package is installed.
3. On VP1 and VP2, verify that the DMFailover-XX.XX.XX.XX-1 package is installed.

Setting Up Host Files or DNS

Create a unique alias for every server with DNS or in the host files to make sure all servers in the failover cluster can connect to each other. The unique alias allows automatic failover reconfiguration without the need of a new IP address when the IP address of another server changes.

1. Define all servers in the cluster by creating a unique alias in the DNS or the host files.
Define the IP addresses in the `/etc/hosts` file if using the host file method.
2. From the original-active and designated-standby servers, add an entry to the `/etc/hosts` to make sure that the local server resolves to a publicly accessible IP address instead of 127.0.0.1.
This allows DSA to run on a remote Data Mover agent and connect to the DSA REST running on the active Data Mover host.

Example: Setting Up Host Files or DNS

1. Define an alias for all five servers in the `/etc/hosts` file on each system (DM1, DM2, DM3, VP1, and VP2), if not already defined.

For example, the following are entries added to the host files on DM1:

- `##.##.###.## DM2`
- `##.##.###.## DM3`
- `##.##.###.## VP1`
- `##.##.###.## VP2`

2. On DM1 and DM2, add a public IP address to the `/etc/hosts` file so that the name of the server does not resolve to the 127.0.0.1 IP address.

An example DM1 entry: `###.##.###.## DM1`

Verifying Required Ports Open

Specific ports must be open on each server in the failover cluster. In most cases, if a default port is unavailable, a different port can be assigned.

1. Use the following chart to verify the required ports are open in the failover cluster:

Port Number	Server Type	Used By
22	Designated-active Designated-standby Standby-only Additional agent Designated-active monitor Designated-standby monitor	SSH
5432	Designated-active	JDBC and Postgres replication

Port Number	Server Type	Used By
	Designated-standby Standby-only	
61616	Designated-active Designated-standby	ActiveMQ
1443	Designated-active Designated-standby	RESTful API
9090	Designated-active Designated-standby	DSA RESTful API

Example: Verifying Required Ports are Open

- Each port only needs to be open for the server listed:

Port Number	Example Server Name	Used By
22	DM1, DM2, DM3, VP1, VP2	SSH
5432	DM1, DM2	JDBC and Postgres replication
61616	DM1, DM2	ActiveMQ
1443	DM1, DM2	RESTful API
9090	DM1, DM2	DSA RESTful API

Defining Unique Data Mover Agent Names

The `agent.id` is set to `Agent1` by default. You must enter a unique name for the Data Mover agents when there are multiple agents for a single Data Mover daemon.

- Enter the unique name for the `agent.id` property in `etc/opt/teradata/datamover/agent.properties` for the designated-active, designated-standby, standby-only, and additional-agent servers.

Example: Defining Unique Data Mover Agent Names

- On DM1, DM2, and DM3, edit the `agent.properties` file and provide a unique name for `agent.id`.

Configuring Cluster Settings

In a high availability configuration, the ActiveMQ broker is enabled on the designated-active Data Mover server. The ActiveMQ broker on the designated-standby server is only activated if a failover occurs.

1. Confirm the network of brokers has been disabled:
`/opt/teradata/client/nn.nn/datamover/failover/dmcluster configactivemq -e false`

Note:

If upgrading from a Data Mover version earlier than 16.20, you must reconfigure the ActiveMQ settings to use only a single active ActiveMQ broker.

2. On both the active and standby Data Mover servers, set the following values for the `daemon.properties` file:

```
cluster.enabled=false
broker.url=localhost or IP address of local system
```

The daemon on the active Data Mover server only uses the local ActiveMQ broker. If failover occurs, the daemon on the standby Data Mover server only uses the ActiveMQ broker on the standby server.

3. On both the active and standby server, and any standalone Data Mover agent servers, set the following values for the `agent.properties` file:

```
cluster.enabled=true
broker.url=active broker host or IP, standby broker host or IP
```

Providing both broker URLs allows the agent to automatically connect to whichever Data Mover daemon is active.

4. On both the active and standby Data Mover server, and any other server where the Data Mover command line is installed, set the following values for the `commandline.properties` file:

```
dm.rest.endpoint=active REST server url, standby REST server url
```

Providing both REST server URLs in `commandline.properties` allows the command line to automatically connect to whichever Data Mover server is active.

Make sure the `host:port` for both the active and standby REST servers are added to the `accept.host.list` in `tdmrest.properties` on both the active and standby Data Mover servers.

Example: Configuring Cluster Settings

1. On both the active and standby Data Mover servers (DM1 and DM2), confirm ActiveMQ is running.

Note:

ActiveMQ on the standby Data Mover server (DM2) is disabled later and should not be running once the configuration steps are completed.

- Run the following command to make sure the network of broker features is disabled:
`/opt/teradata/client/nn.nn/datamover/failover/dmcluster configactivemq -e false`
- On both the active and standby Data Mover servers (DM1 and DM2), set the following in the `daemon.properties` file:

```
cluster.enabled=false
broker.url=localhost or IP address of local system
```

- On the active, standby, and any standalone Data Mover servers (DM1, DM2, and DM3), set the following in the `agent.properties` file:

```
cluster.enabled=true
broker.url=DM1,DM2
```

- On both the active and standby Data Mover servers (DM1 and DM2) and any standalone Data Mover servers, set the following in the `commandline.properties` file:

```
dm.rest.endpoint=DM1 REST URL,DM2 REST URL
```

Configuring the Sync Service Properties

- On the designated-active server, edit the `sync.properties` file:

Property	Value
<code>sync.isMaster</code>	True

- On the designated-standby and standby-only servers, edit the `sync.properties` file:

Property	Value
<code>sync.isMaster</code>	False

- On the Data Mover designated-active repository Postgres configuration, change the **wal_level** value to logical:

```
/var/opt/teradata/postgres/data/postgres.conf: wal_level = logical
```

- Restart the Data Mover repository:

```
/etc/init.d/postgresql stop
/etc/init.d/postgresql start
```

- Restart DM services using the command `/opt/teradata/datamover/daemon/nn.nn/dm-control.sh restart`, where nn.nn refers to the version numbers of Data Mover.

Note:

If the DM services do not restart, the initial datamove commands fail but the subsequent commands continue to work.

Example: Configuring the Sync Service

- On DM1, enter the following changes to the `sync.properties` file:

Property	Value
<code>sync.isMaster</code>	<code>True</code>

- On DM2, enter the following changes to the `sync.properties` file:

Property	Value
<code>sync.isMaster</code>	<code>False</code>

- On DM1, change the Data Mover repository Postgres configuration.
`/var/opt/teradata/postgres/data/postgres.conf: wal_level = logical`
- On DM1, restart the repository.
`/etc/init.d/postgresql stop`
`/etc/init.d/postgresql start`
- Restart DM services using the command `/opt/teradata/datamover/daemon/nn.nn/dm-control.sh restart`, where nn.nn refers to the version numbers of Data Mover.

Note:

If the DM services do not restart, the initial datamove commands fail but the subsequent commands continue to work.

Configuring the Cluster and Starting the Monitoring Service

The monitoring service uses SSH connections to the servers it monitors and JDBC to monitor the Data Mover repositories. The command can take a few minutes to complete and does the following:

- Sets up the SSH logons for the monitoring services so that the services can log on without a password to the monitored servers.
- Sets up and stores the usernames and passwords for the designated-active and standby repositories.

- Sets up ActiveMQ authentication for the designated-active and standby ActiveMQ, daemon, agents, and REST servers.
 - Configures and activates the synchronization service from the designated-active to the designated-standby repository.
 - Configures and starts the designated-active daemon, agents, REST, bundled DSA, and designated-active sync monitor service in active mode.
 - Starts the monitoring service on `local.monitor.host` to monitor the local Data Mover components.
1. Log on to the designated-active daemon server and edit the `/etc/opt/teradata/datamover/failover.properties` file for your system.
For details about the `Failover.properties` file, see [Failover.properties File](#)
 2. Run the following command as root:
`./dmcluster config`

Note:

In a default installation, the active repository host is the same as the active daemon host and the standby repository host is the same as the standby daemon host.

Failover.properties File

The Data Mover `failover.properties` file contains the files that control the failover process. When setting up the failover process, edit these files according to the system being configured.

Property Name	Description
<code>local.daemon.host</code>	Host where the designated-active daemon runs.
<code>remote.daemon.host</code>	Host where the designated-standby daemon runs.
<code>local.monitor.host</code>	Host where the monitoring service that monitors the designated-active services runs.
<code>remote.monitor.host</code>	Host where the monitoring service that monitors the designated-standby services runs.
<code>local.repository.host</code>	Host where the repository used by the designated-active daemon is installed. Use the same host the sync monitor service is installed on. If the repository is installed on the same server as the daemon, this value is the same as <code>local.daemon.host</code> .
<code>remote.repository.host</code>	Host where the repository used by the designated-standby daemon is installed. Use the same host the sync monitor service is installed on. If the repository is installed on the same server as the daemon, this value is the same as <code>remote.daemon.host</code> .

Property Name	Description
<code>local.agents.host</code>	Host where the agents used by the designated-active daemon are installed. If more than one agent is used, use a comma-separated list to specify agents; the order of the list does not matter.
<code>remote.agents.host</code>	The hosts where the agents used by the designated-standby daemon are installed. If more than one agent is used, use a comma-separated list to specify agents; the order of the list does not matter.

If external agents are shared between the active and standby server, the shared agent names must be specified for both `local.agents.host` and `remote.agents.host`.

Example: Configuring the Cluster and Starting the Monitoring Service

The `dmcluster config` command sets up SSH connections between the DM1, DM2, DM3, VP1, and VP2 servers. These connections are used by the Data Mover monitoring service to monitor the designated-active Data Mover components and initiate a failover if any of the designated-active components stop working.

1. On the designated-active (DM1) server, enter the following changes to the `failover.properties` file:

- `local.daemon.host=DM1`
- `remote.daemon.host=DM2`
- `local.monitor.host=VP1`
- `remote.monitor.host=VP2`
- `local.repository.host=DM1`
- `remote.repository.host=DM2`
- `local.agents.host=DM1, DM2, DM3`
- `remote.agents.host=DM1, DM2, DM3`

2. Run `./dmcluster config` as root and follow the following prompts:

Prompt	Action
Please enter the username for the ACTIVE repository [datamover]:	Provide the database user name for the designated-active Data Mover repository on DM1.
Please enter the password for ACTIVE repository:	Provide the password for the mentioned username on the designated-active Data Mover repository on DM1.
Please enter 'postgres' user password for ACTIVE repository:	Provide the Postgres user password for the designated-active Data Mover repository on DM1.

Prompt	Action
Please enter the username for the STANDBY repository [datamover]:	Provide the database username for the designated-standby Data Mover repository on DM2.
Please enter the password for STANDBY repository:	Provide the password for the mentioned username on the designated-standby Data Mover repository on DM2.
Please enter 'postgres' user password for STANDBY repository:	Provide the Postgres user password for the designated-standby Data Mover repository on DM2.
Please enter the Root Password for the server running ACTIVE Repository:	Provide the root Linux password for DM1.
Please enter the Root Password for the server running STANDBY Repository:	Provide the root Linux password for DM2.
Please enter the password for root@DM1(ACTIVE DAEMON):	Provide the root Linux password for DM1.
Please enter the password for root@DM2(STANDBY DAEMON):	Provide the root Linux password for DM2.
Please enter the password for root@VP1(ACTIVE MONITOR):	Provide the root Linux password for VP1.
Please enter the password for root@VP2(STANDBY MONITOR):	Provide the root Linux password for VP2.
Please enter the password for root@DM3(ACTIVE Agent(s)):	Provide the root Linux password for DM3.
Do you need to backup the ACTIVE repository to the STANDBY repository? [y or n]	Failover requires that the designated-active and designated-standby repositories are in sync. Enter y to sync up the repositories.

Checking the Status of Active and Standby Components

1. Verify the status of the cluster:
./dmcluster status

Example: Checking the Status of the Active and Standby Components

1. View the newly configured cluster status:
./dmcluster status

It may take a few minutes for all services to transition. Run `dmcluster status` as needed until the results are as expected. Monitor progress in `dmFailover.log` on the currently active daemon and monitor servers.

LOCAL CLUSTER		
COMPONENT	HOST NAME	STATUS
DM Daemon	DM1	RUNNING
ActiveMQ	DM1	RUNNING
DM Monitoring Service	VP1	RUNNING
DM Sync Monitor	DM1	RUNNING
DM REST Service	DM1	RUNNING
DSC	DM1	RUNNING
DSA REST	DM1	RUNNING
DM Agent	DM1	RUNNING
DM Agent	DM2	RUNNING
DM Agent	DM3	RUNNING
REMOTE CLUSTER		
COMPONENT	HOST NAME	STATUS
DM Daemon	DM2	STOPPED
ActiveMQ	DM2	STOPPED
DM Monitoring Service	VP2	STOPPED
DM Sync Monitor	DM2	STOPPED
DM REST Service	DM2	STOPPED
DSC	DM2	STOPPED
DSA REST	DM2	STOPPED
DM Agent	DM2	STOPPED
DM Agent	DM1	RUNNING
DM Agent	DM2	RUNNING
DM Agent	DM3	RUNNING

Cluster Mode: DM Cluster is in 'Active Mode'

In the example output, only the Data Mover sync monitor on the active system is running to monitor synchronization progress from the designated-active to the designated-standby repositories. The sync monitor on the standby system is never used.

Note:

If you configure Data Mover to use DSC on its host, the status for DSC and DSA REST is either RUNNING or STOPPED. If you configured Data Mover to use an external DSC, the status for DSC and DSA REST is EXTERNAL. See [DSA Failover Configurations](#) for more details.

Verifying Failover Configuration

1. On the designated-active server, verify that all Data Mover agents are listed in the output:
`datamove list_agents`
2. On the designated-standby, verify the daemon is not running:
`/etc/init.d/dmdaemon status`
3. On the designated-standby, verify the REST service is not running:

```
/etc/init.d/tdmrest status
```

4. On the designated-standby, verify the ActiveMQ is not running:

```
/etc/init.d/tdactivemq status
```
5. Run a Data Mover TPT job.
 - a. Set `max_agents_per_task` to the number of Data Mover agents in the cluster.
 - b. Set `data_streams` to a value greater than or equal to `max_agents_per_task`.
 - c. Check the job status to verify that the job ran successfully and that all Data Mover agents are in use.

Example: Verifying the Failover Configuration

1. On DM1, run `datamove list_agents` and verify that all three Data Mover agents (DM1, DM2, and DM3) are running.
2. On DM2, run `/etc/init.d/dmdaemon status` and verify that the output indicates NOT RUNNING.
3. On DM2, run `/etc/init.d/tdmrest status` and verify that the output is NOT RUNNING.
4. On DM2, run `/etc/init.d/tdactivemq status` and verify that the output status is NOT RUNNING.
5. Run a Data Mover TPT job.
 - a. Set `max_agents_per_task` to the number of Data Mover agents in the cluster.
 - b. Set `data_streams` to a value greater than or equal to `max_agents_per_task`.
 - c. Check the job status to verify that the job ran successfully and that all Data Mover agents are in use.

Completing the Automatic Failover Setup

1. Run `./dmcluster status` to check the configuration status.

Simulating a Failover Event and Restoring Cluster

Do the following to simulate a failover event and then restore the cluster to the original configuration:

Simulating a Failover Event

1. View the current cluster status: `./dmcluster status`
2. On the designated-active system, shut down the Data Mover repository: `/etc/init.d/postgresql stop`
3. Check the cluster status to verify the initiation of failover: `./dmcluster status`

Note:

The failover monitor recognizes that the Data Mover repository on the designated-active system is down. However, the failover monitor makes retry attempts to check the repository status before initiating failover. So the process can take time.

Note:

Stopping the Data Mover services, such as the daemon or agent, does not result in a failover event. The failover monitor instead detects that those services are down and attempts to restart the services before initiating a failover.

Restoring the Cluster to the Original Configuration:

4. On the designated-active system, restart the Data Mover repository: `/etc/init.d/postgresql stop`
5. On the designated-active system, run the command: `./dmcluster switchback`
6. Verify the restoration of the original cluster configuration: `./dmcluster status`

DSA Failover Configurations

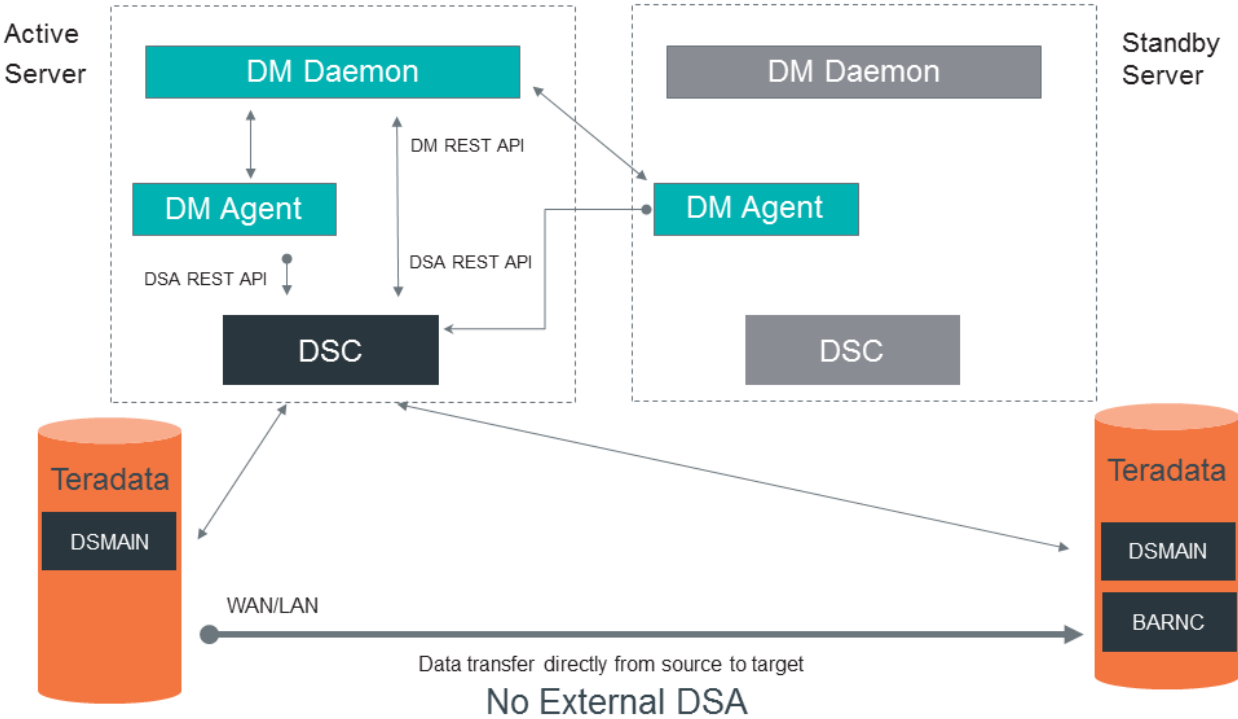
How you configure failover with DSA depends on which configuration is used for the DSA environment. For a description of the configurations, see [DSA Configurations](#).

Note:

DSA does not support automatic failover. If using the embedded Data Mover DSC, additional configuration may be needed after failover occurs before Data Mover DSA jobs can be run again.

Configuration 1: DSC Running on the Data Mover Server

When there is no external DSA environment, the designated-active and designated-standby Data Mover daemons are configured with the bundled DSC provided in the Data Mover package.



In this configuration, DSC, DSA REST, and Teradata ActiveMQ are running on the designated-active Data Mover server. On the designated-standby server, these components are turned off. If failover occurs, the Data Mover failover monitor automatically turns on the Data Mover components on the designated-standby server, including DSC, DSA REST, and Teradata ActiveMQ. However, additional configuration is required before Data Mover DSA jobs can be run on the new active Data Mover server.

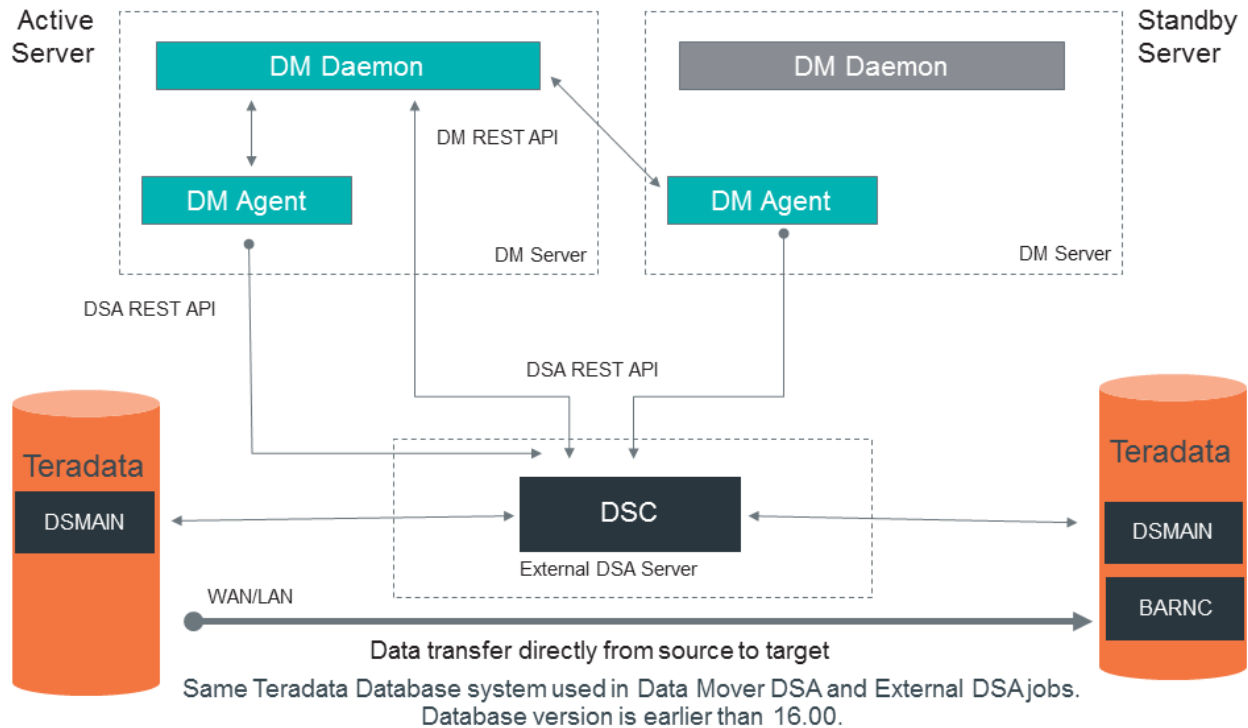
Refer to the following considerations based on what version of the Teradata system is running on the source and target systems:

Teradata Version	Action Needed	Description
Earlier than 16.00	Yes	Before running any Data Mover DSA jobs, you must go through all of the steps to configure the DSC on the new active Data Mover server. No pre-configuration can occur in this scenario.
16.00 and later	No	The DSC on the designated-active and designated-standby servers are registered with the source and target systems during the configuration process, and the DSA ClientHandler is pre-configured to use the designated-standby DSC. If failover occurs, no additional action is needed.

Configuration 2: Data Mover using an External DSC

Configuration 2 is used when a Teradata Database with a version earlier than 16.00 has an existing DSA environment and is used for the following:

- Backup and restore
- Copying data with Data Mover using the database as a source or target system



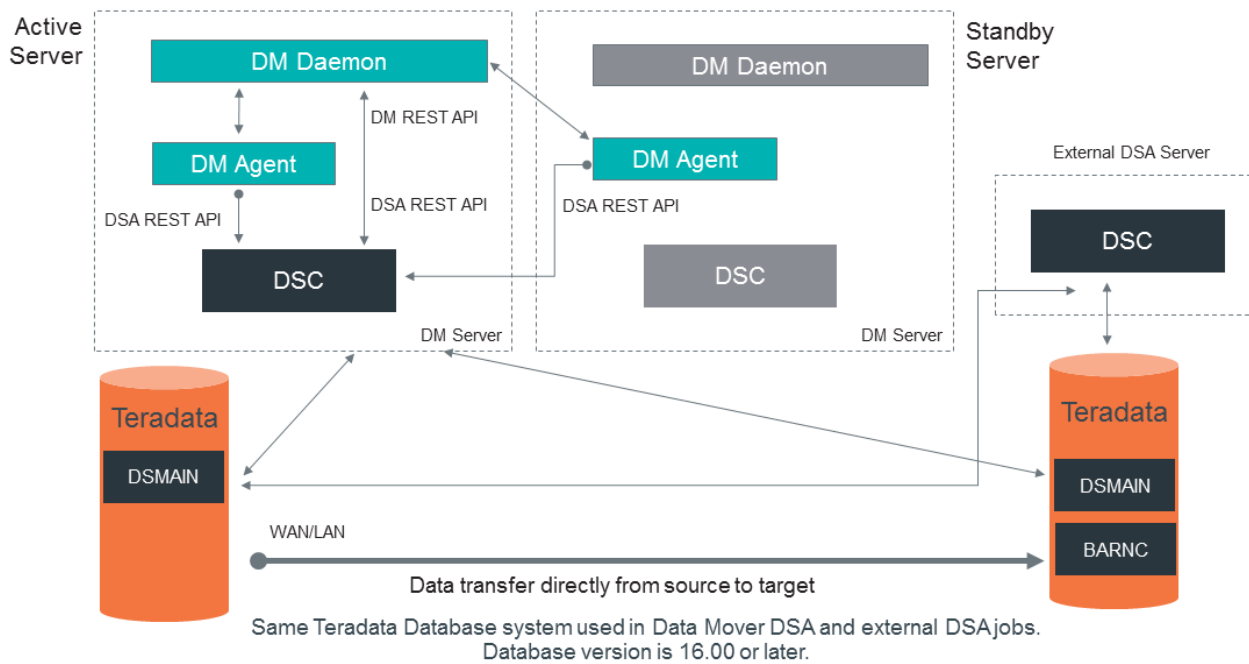
In this configuration, both the designated-active and designated-standby servers use the same external DSC. If failover occurs, the failover monitor starts the designated-standby Data Mover components, which then calls the external DSC. No DSC configuration or registration is necessary if a Data Mover failover occurs.

Note:

If the external DSC fails, you need to [register a new external DSC](#) with the source and target systems and configure the designated-active and designated-standby Data Mover servers to use this new DSC. There is currently no automatic failover for this type of failure.

Configuration 3: DSC running on the Data Mover server and externally

Configuration 3 is when you use both the Data Mover DSC and an external DSC environment on Teradata Databases 16.00 and later.



For failover, this configuration mirrors Configuration 1. The independent, external DSC is not involved in Data Mover failover scenarios. See DSA Failover Configuration 1 for details on the bundled DSC actions.

Configuring Failover for DSC Running on the Data Mover Server

Refer to the following failover configuration steps when running DSA on the Data Mover server. These steps can be performed prior to a failover event and apply to systems using the following specifications:

- Data Mover 16.20 or later
- Teradata system versions are 16.00 or later on both the source and target

1. Stop the designated-active monitor:

```
/opt/teradata/client/nn.nn/datamover/failover/dmcluster stopmonitor
```

2. On the designated-standby host, start Teradata ActiveMQ and DSC:

```
/etc/init.d/tdactivemq start
/etc/init.d/dsc start
```

3. See [Configuring the Source and Target Systems](#) section for information on registering the source and target systems with DSC and restart DSMAN.
4. Perform the following steps for the source and target TPA nodes with BAR NC ClientHandler software installed:
 - a. Append the designated-standby Data Mover host to `broker.list` found in the `/etc/opt/teradata/dsa/clienthandler.properties` file as in the following example:

```
broker.list=153.64.24.162:61616,153.64.24.164:61616
```

If upgrading, refer to the saved `clienthandler.properties` file created during the upgrade.

- b. Restart the ClientHandlers.

```
/etc/init.d/clienthandler restart
```

5. Configure the DSA network, see [DSA Network Configuration](#).
6. On the designated-standby host, stop the DSC and `tdactivemq`:

```
/etc/init.d/dsc stop
/etc/init.d/tdactivemq stop
```

7. Start the designated-active monitor.
8. Run `dmcluster status` and verify that the DSC and DSA REST components are marked as RUNNING on the designated-active Data Mover server and that the DSC and DSA REST components are marked as STOPPED on the designated-standby Data Mover server.

Note:

If failover occurs, the Data Mover failover monitor automatically fails over to the designated-standby server. Additional configuration is required to enable Data Mover DSA jobs on the designated-standby server. Refer to the tasks:

- [Configuring DSA after Failover Occurs](#)
 - [Reverting Back to the Designated-Active Server](#)
-

Configuring DSA after Failover Occurs

If failover occurs, the Data Mover failover monitor automatically fails over to the standby server.

1. Perform one of the following actions, based on the version of Teradata on your source and target systems:

Teradata Version	Action
16.00 or later	No additional steps are necessary.
Earlier than 16.00	<ul style="list-style-type: none"> • Run <code>dsa_configsys</code> from the new active Data Mover server to register the system • Restart DSMAIN. • Follow the configuration steps, based on your environment, in the DSA Network Configuration section.

Reverting Back to the Designated-Active Server

After a failover, perform the following steps after restoring the primary or designated active Data Mover server to revert it back to an active state.

1. Return the primary active server to an active state.
2. Follow the steps as listed in [Configuring DSA after Failover Occurs](#).

Enabling TLS 1.2 Data Path Encryption for Standby DSC

Prerequisite:

TLS 1.2 Data Path encryption is enabled on the Active DSC. See [Enabling TLS 1.2 Data Path Encryption for DSC on the Data Mover Server](#)

(Optional configuration). Perform the following steps:

1. Add TLS related properties in `dsc.properties` similar to that of the active DSC
2. Restart DSC.
3. Re-configure source system and restart `dsmain`, using `dsa_configsys` or `dsc commandline`
4. Re-configure target system and restart `dsmain`, using `dsa_configsys` or `dsc commandline`

Configuring Failover for Data Mover Using an External DSC

Refer to the following failover configuration steps when running DSA on an external server.

1. Configure the standby Data Mover daemon to use the external DSC by editing `/etc/opt/teradata/datamover/daemon.properties` with the following information:

```
dsa.rest.endpoint=http://external_dsc_host:9090/dsa
is.dsc.colocate.dm=false
```

Note:

If the external DSA REST server is configured with HTTPS, replace `http` with `https` in the `dsa.rest.endpoint` codeblock.

2. Stop the DSC on the standby host using the `/etc/init.d/dsc stop` command.
3. Run `dmcluster status` and verify that the DSC and DSA REST components are marked EXTERNAL on both the active and standby Data Mover hosts.

Note:

If the DSC or DSA REST component is marked RUNNING or STOPPED, verify that the `is.dsc.colocate.dm` property is set in the `daemon.properties` file as false.

Note:

If Data Mover failover occurs, the Data Mover failover monitor automatically fails over to the standby Data Mover server. If the external DSA fails, both active and standby Data Mover servers need to be reconfigured to use the new DSC. See [Configuring Data Mover to Use an External DSC](#).

Configuring the Synchronization Service without Automatic Failover

Configure the synchronization service when automatic failover cannot be used. The synchronization service uses the following files:

File	Description
<code>/etc/opt/teradata/datamover/sync.properties</code>	Data Mover Replication service settings used for synchronizing the active with the standby repositories.
<code>/opt/teradata/datamover/sync/nn.nn/DMReplication.jar</code>	Executable binary file used by the synchronization service.
<code>/opt/teradata/datamover/sync/nn.nn/dmsync</code>	Script for configuring the synchronization service.

Where `nn.nn` in the path refers to the version numbers of Data Mover.

Configuring the Synchronization Service

When configuring the sync service, do not have running jobs on either the active or standby servers and make sure the repositories are in synchronization. You can synchronize the repositories using one of the following methods:

- Before configuration: Run the `backup_daemon` and `restore_daemon` commands
- During configuration: Let `dmsync config` perform the backup and restore during the configuration process

Note:

The `dmsync config` command does not check if there are running jobs when doing a backup and restore of the system. If this is a concern, run the Data Mover backup and restore commands before configuring the synchronization service.

1. On the active repository server, go to `cd /var/opt/teradata/postgres/data` to change the Postgres configuration.
2. In `postgresql.conf`, change `wal_level` to `logical`.
`wal_level = logical`
3. Restart the active repository:
 - `/etc/init.d/postgresql stop`
 - `/etc/init.d/postgresql start`
4. On the active server, edit the `sync.properties` file and set the `sync.isMaster` property to `true`.
5. On the standby server, edit the `sync.properties` file and set the `sync.isMaster` property to `false`.
6. On the active server, set the `dm.pg.jobstore.production.host` property in the `sync.properties` file as the hostname for the active server.
Use a name other than `localhost` as the host name.
7. On the active server, configure the sync service:
`/opt/teradata/datamover/sync/nn.nn/dmsync config`
Where `nn.nn` in the path refers to the version numbers of Data Mover.
8. When prompted, respond to the following:
 - a. Select the **Setup Replication Service** option.
 - b. Provide the standby server name.
 - c. On the standby server, provide the Postgres user credentials.
 - d. Answer `y` or `n` to backup and restore the repository from the active to the standby server.
9. Verify that `/var/opt/teradata/logs/dmSync.log` is configured properly and the sync monitor is running on the active server.
The monitoring service automatically starts once the configuration process completes successfully.

Dropping Existing Synchronization Service

1. From the active sync server, drop the current synchronization service configuration:
`/opt/teradata/datamover/sync/nn.nn/dmsync dropconfig`
Where `nn.nn` in the path refers to the version numbers of Data Mover.
2. When prompted, respond to the following:
 - a. Select the **drop Replication Service** option.
 - b. Provide the standby server name.
 - c. Provide the Postgres user credentials on the standby server.

- d. Answer y or n to drop more standby servers.

Starting or Stopping the Synchronization Monitor Service

The synchronization monitoring service automatically starts running after configuring the synchronization service. In cases where you need to manually start or stop the service, use the following step:

1. From the active sync server, start or stop the synchronization monitor service:

```
/opt/teradata/datamover/sync/nn.nn/dmsync startmonitor
```

```
/opt/teradata/datamover/sync/nn.nn/dmsync stopmonitor
```

Where *nn.nn* in the path refers to the version numbers of Data Mover.

Remove High Availability from a Cluster

Use the following instructions to remove a high availability environment from a cluster. Mention of *nn.nn* in any filepath refers to the major and minor version numbers of Data Mover.

1. From the active daemon in the `/opt/teradata/client/nn.nn/datamover/failover/` directory, run the following command:


```
./dmcluster configactivemq -e false
```
2. From the standby daemon in the `/opt/teradata/client/nn.nn/datamover/failover/` directory, run the following command:


```
./dmcluster configactivemq -e false
```
3. On the active daemon, find the active monitor server by running the following command:


```
./dmcluster status
```
4. From the active monitor server in the `/opt/teradata/client/nn.nn/datamover/failover/` directory, run the following command:


```
./dmcluster stopmonitor
```
5. On both the active and standby daemons in the `/etc/opt/teradata/datamover` directory, edit the `daemon.properties`, `tdmrest.properties`, and `agent.properties` files with the following values:
 - `cluster.enabled = false`
 - `broker.url = localhost`
6. Edit the `commandline.properties` file with the following values:


```
dm.rest.endpoint=https://localhost:1443/datamover
```
7. From the `/opt/teradata/datamover/sync/nn.nn/` directory, run `./dmsync dropconfig` to remove the synchronization service from the active to standby repository, see [Dropping Existing Synchronization Service](#).
8. From the active daemon, edit the following properties in the `failover.properties` file as `localhost` in the `/etc/opt/teradata/datamover` directory:
 - `local.daemon.host`
 - `remote.daemon.host`

- local.monitor.host
 - local.repository.host
 - remote.repository.host
 - local.agents.host
 - remote.agents.host
9. On both the active and standby daemon, restart the tdactivemq, daemon, REST, and agent services.


```

/etc/init.d/tdactivemq stop
/etc/init.d/tdactivemq start
/etc/init.d/dmdaemon stop
/etc/init.d/dmdaemon start
/etc/init.d/dmagent stop
/etc/init.d/dmagent start
/etc/init.d/tdmrest stop
/etc/init.d/tdmrest start
      
```
 10. On both the active and standby daemon, run the `datamover list_agents` command to verify the daemon is connected to ActiveMQ and running on localhost.

Setting up a TLS 1.2 Connection for Replication Service

Perform the following steps to set up TLS 1.2 for the PostgreSQL Replication Service using your own certificates:

1. If active, stop the failover monitoring service:


```

/opt/teradata/client/nn.nn/datamover/failover/dmcluster stopmonitor
      
```

Where *nn.nn* in the path refers to the version numbers of Data Mover.
2. Run the following on both the primary and secondary systems:
 - a. Edit the `/var/opt/teradata/postgres/data/postgresql.conf` configuration file.
 - b. Replace the following properties with your certificate files:
 - `ssl_cert_file=server certificate`
 - `ssl_key_file=server private key`
 - `ssl_ca_file=trusted certificate authorities`

For more information on these properties, refer to <https://www.postgresql.org/docs/10/runtime-config-connection.html#GUC-SSL-CERT-FILE>.

- c. Stop the Daemon service:


```

/etc/init.d/dmdaemon stop
      
```
- d. Stop the DSC service:


```

/etc/init.d/dsc stop
      
```
- e. Restart the Postgres service:


```

/etc/init.d/postgresql restart
      
```
- f. Start the DSC service:


```

/etc/init.d/dsc start
      
```

- g. Start the Daemon service:

```
/etc/init.d/dmdaemon start
```

3. Start the failover monitoring service if it was previously configured:

```
/opt/teradata/client/nn.nn/datamover/failover/dmcluster startmonitor
```

Configuring Data Mover to Use Teradata Ecosystem Manager

1. In the `/etc/opt/teradata/datamover/apiconfig.xml` file, edit the host and port properties for the location of the Resilient Publisher.
2. Run the `list_configuration` command to output a configuration file.
3. Set the appropriate values for the configuration settings for your site.

Parameter	Description	Default
tmsm.frequency.bytes	Controls the frequency, in number of bytes/MB/GB, of job progress events sent to Teradata Ecosystem Manager. Note: Using a low value can hurt performance. The recommendation is to use the default value.	2147483647
tmsm.mode	Controls how Data Mover directs Teradata Ecosystem Manager messages. Valid Values: <ul style="list-style-type: none"> • BOTH • ONLY_REAL_TMSM • ONLY_INTERNAL_TMSM • NONE When set to: <ul style="list-style-type: none"> • BOTH, messages are sent to the Teradata Ecosystem Manager system and written to the table-driven interface event tables. • ONLY_INTERNAL_TMSM, Data Mover only writes messages to the TMSMEVENT table defined by the table-driven interface. • ONLY_REAL_TMSM, Data Mover only sends messages to the Teradata Ecosystem Manager system. If Data Mover cannot send events to the real Teradata Ecosystem Manager product then those events will be stored in a <code>store.dat</code> file located in the <code>opt/teradata/client/em/dataStore/store.dat</code> directory. If the value for <code>tmsm.mode</code> is BOTH or ONLY_REAL_TMSM, and Data Mover cannot send events to the real Teradata Ecosystem Manager product, then the <code>store.dat</code> file can grow to be very large. To prevent the <code>store.dat</code> file from taking up too much disk space on the Data Mover multi-purpose server, change the value for <code>tmsm.mode</code> to ONLY_INTERNAL_TMSM or NONE, or make sure Data Mover can send events to the real Teradata Ecosystem Manager product.	NONE

For more information about Teradata Ecosystem Manager, see the *Teradata® Ecosystem Manager User Guide*.

Configuring Multiple Multi-Purpose Servers

Having more than one Data Mover multi-purpose server in the environment can improve performance when copying data from one Teradata Database system to another. Each Data Mover multi-purpose server can have one or more Data Mover components running on it.

If the Data Mover agent must be run on a system other than the Data Mover daemon, the host name for the server running the Data Mover daemon must be resolved to a publicly-accessible IP address in the `/etc/hosts` file.

If only agents are running on the additional Data Mover multi-purpose servers, they must be configured to work with the Data Mover multi-purpose server that has the Data Mover daemon running on it.

When using multiple Data Mover agents, each Data Mover agent must have a unique Agent ID.

1. Provide the correct Teradata ActiveMQ broker `url` and `port` number values in one of the following ways:
 - During installation of the Data Mover agent component on the Data Mover multi-purpose server
 - After installation by modifying the `broker.url` and `broker.port` in the `agent.properties` file where ActiveMQ runs.
2. Edit the `Agent ID` property in the `agent.properties` file.
3. Restart the Data Mover agent service to implement the changes.

Configuring Data Mover to Log to Server Management

Configure Data Mover to log to the Server Management by using the `tvi.useLogger` property in the `agent.properties` and `daemon.properties` files. The property defaults to `True`, making Server Management logging automatic and enabling critical failures to be immediately reported to Teradata.

TVI alerts are only sent when CMIC is configured.

Note:

Data Mover components load CMIC configuration properties in `sm_config.txt` on startup only. Restart Data Mover components to pick up changes in `sm_config.txt` after configuration.

1. Log on to the agent and daemon servers.
2. Set the `tvi.useLogger` property in the `daemon.properties` file and the `tvi.useLogger` property in the `agent.properties` file to `True`.

Enabling Logging Server Management Alerts When a Failover Occurs

1. Log on to the active and standby monitoring servers: `local.monitor.host` and `remote.monitor.host`.
2. On `/opt/teradata/client/nn.nn/datamover/failover/monitor.properties`, do the following:
 - Set the value for **monitor.useTviLogger** as `True`.
 - Make sure that `tvilogger.properties` exists and has been configured with the correct Server Management logging method.

Configuring Data Mover Multi-Purpose Server to Increase Network Throughput

All network traffic coming into and out of the Data Mover multi-purpose server goes through the default Ethernet port for the server unless it is specifically routed. If the default Ethernet port is used for all network communication, the other network ports on the Data Mover multi-purpose server are wasted. This could cause the network to slow down when processing Data Mover jobs, which could lead to poor performance when copying data.

Data Mover jobs execute much faster if multiple Ethernet ports are used when copying data between Teradata Database systems. The recommended way to increase network throughput on the Data Mover multi-purpose server is to set up specific network routes for all of the COP entries on the source and target Teradata Database systems in the Data Mover jobs. A COP entry is the IP address of a Teradata Database node. These specific network routes allow the Data Mover Agent to connect TCP sessions to the source and target systems using different Ethernet ports on the Data Mover multi-purpose server. This improves performance by distributing data across all available network ports.

The topics in this section describe how to set up the routes using a 2-node Teradata Database system called `dmdev` as a source and a 2-node Teradata Database system called `dmtest` as target. The examples in this section assume the network ports `eth4` and `eth5` are connected and available for use on the Data Mover multi-purpose server.

Note:

More than two ports on the Data Mover multi-purpose server could be available in a customer environment. The examples in this section use only 2-node source and target systems and two available network ports on the Data Mover multi-purpose server.

1. Add the IP addresses for all source and target COP entries in the `/etc/hosts` file on the Data Mover multi-purpose server.
2. Define the specific routes for the COP entries in the `/etc/sysconfig/network/routes` file on the Data Mover multi-purpose server.

3. Restart the network on the Data Mover multi-purpose server.
4. Verify the route changes are in place on the Data Mover multi-purpose server.

Adding Source and Target COP Entries

The best way to define the IP addresses for the source and target COP entries is to configure them through DNS. The example here defines the IP addresses for the source and target COP entries in the `/etc/hosts` file instead because it is easier to explain all of the steps this way.

The IP addresses (COP entries) for all nodes on the source and target systems are placed in the `/etc/hosts` file so the Data Mover agent can resolve them when running a job. As an example, if the IP addresses of the two nodes on `dmdev` are 153.64.209.91 and 153.64.209.92, respectively, and the IP addresses of the two nodes on `dmtest` are 153.64.106.78 and 153.64.106.79, respectively, we add the following entries to the `/etc/hosts` file on the Data Mover multi-purpose server:

```
# COP entries for dmdev
153.64.209.91 dmdev dmdevcop1
153.64.209.92 dmdev dmdevcop2

# COP entries for dmtest
153.64.106.78 dmtest dmtestcop1
153.64.106.79 dmtest dmtestcop2
```

The COP entries for the source and target systems are now in the `/etc/hosts` file.

Defining Routes for Source and Target COP Entries

Next, the network routes for the COP entries can be added to the `/etc/sysconfig/network/routes` file. Assume the `eth2` interface is used for all public network traffic to and from the Data Mover multi-purpose server and is, therefore, the default network interface for the server. Assume the IP address 153.64.107.254 is the gateway for all traffic coming into and out of the Data Mover multi-purpose server. The following is added to the `/etc/sysconfig/network/routes` file on the Data Mover multi-purpose server to add specific routes for the COP entries on `dmdev` and `dmtest`:

```
# default XXX.XXX.XXX.XXX - ethX
default 153.64.107.254 - eth2

# routes to system dmdev
153.64.209.91 153.64.107.254 - eth4
153.64.209.92 153.64.107.254 - eth4

# routes to system dmtest
153.64.106.78 153.64.107.254 - eth5
153.64.106.79 153.64.107.254 - eth5
```


These entries force all network traffic between the Data Mover multi-purpose server and dmdev to use the eth4 interface and all network traffic between the Data Mover multi-purpose server and dmttest to use the eth5 interface.

Restarting the Network

The network on the Data Mover multi-purpose server must be restarted for the changes in the `/etc/sysconfig/network/routes` file to take effect.



NOTICE

Be sure to check that restarting the network will not negatively affect any other users on the server prior to executing this command.

1. Run the `rcnetwork restart` command to restart the network on the Data Mover multi-purpose server.

Verifying the Route Changes

The new routes configured can be verified with the `ip` or `netstat` commands. Following are example outputs of these commands when the routes have been configured properly:

```
# ip route list
153.64.209.92 via 153.64.107.254 dev eth4
153.64.106.78 via 153.64.107.254 dev eth5
153.64.106.79 via 153.64.107.254 dev eth5
153.64.209.91 via 153.64.107.254 dev eth4
127.0.0.0/8 dev lo scope link
default via 153.64.107.254 dev eth2

# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
153.64.209.92 153.64.107.254 255.255.255.255 UGH 0 0 0 eth4
153.64.106.78 153.64.107.254 255.255.255.255 UGH 0 0 0 eth5
153.64.106.79 153.64.107.254 255.255.255.255 UGH 0 0 0 eth5
153.64.209.91 153.64.107.254 255.255.255.255 UGH 0 0 0 eth4
127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo
0.0.0.0 153.64.107.254 0.0.0.0 UG 0 0 0 eth2
```

Data Mover Log Files

Data Mover log files are moved to the `/var/opt/teradata/datamover/logs` directory as follows:

- `dmDaemon.log`
- `dmAgent.log`
- `dmSync.log`
- `dmFailover.log`
- `upgrade_backup.log`

During Data Mover installation and upgrade, the log files are preserved with up to 10 backups. For example, `dmdaemon-postinstall.log` backups are preserved as `dm-daemon-postinstall.log.1`, `dmdaemon-postinstall.log.2`, and so on, up to `dmdaemon-postinstall.log10`, where the most recent file is `dmdaemon-postinstall.log.1`. The following log files, with date and timestamp details added, are preserved during installations and upgrades:

- `/tmp/dmdaemon-postinstall.log`
- `/tmp/dmagent-postinstall.log`
- `/tmp/put-dmschemaupgrade.log`
- `/tmp/put-dmlistagents.log`
- `/var/opt/teradata/datamover/logs/upgrade_backup.log`

Log files for DSA are located in the `/var/opt/teradata/dsa/logs` directory.

Log files for the Data Mover RESTful API are located in the `/var/opt/teradata/datamover/logs` directory as `tdmRest.log` file.

The log file `/tmp/dmdaemon-preupgradecheck.log` captures the output of the Data Mover pre-upgrade check.

Data Mover Properties Files Preserved During Upgrades

For DataMover 16.00.00.00 and later, settings from the following properties files are preserved during upgrades:

- `/etc/opt/teradata/datamover/agent.properties`
- `/etc/opt/teradata/datamover/commandline.properties`
- `/etc/opt/teradata/datamover/daemon.properties`
- `/etc/opt/teradata/datamover/failover.properties`
- `/etc/opt/teradata/datamover/monitor.properties`
- `/etc/opt/teradata/datamover/sync.properties`
- `/etc/opt/teradata/datamover/tdmrest.properties`

DSA Setup for New Teradata Systems

When new databases or systems are added to the environment, you must install the ClientHandler on all TPA nodes. Both the DSA ClientHandler and DSC must be version 16.20 or later. The ClientHandler version can be an exact matching version to the DSC version or three maintenance versions back. After Installing Data Mover, install the ClientHandler using the following steps:

- [Configuring the Source and Target Systems](#)
- [Installing and Configuring BAR NC on TPA Nodes](#)
- Configure a [Netmask](#) or a [Network Fabric](#)

Configuring the Source and Target Systems

Use the `dsa_configsys` tool to configure source and target Teradata systems when using Data Mover with DSA. When the `-h` and `-u` options are provided in the configuration, `dsa_configsys` restarts DSMAIN on the source and target systems. If the `-h` and `-u` options are not provided, DSMAIN must be manually restarted using `cnstern` before using DSA.

The following must be considered when configuring the source and target systems:

- The DSC host must resolve to an IP that can be reached by every TPA node on the source and target systems.
- Each TPA node must resolve COP entries for all nodes on the same system.
- The user specified with the `-u` option must be either "root" or a user with **sudo** access to run the Teradata `cnstern` utility.
- You must specify TDPID with the `-t` option.

Note:

`build_dsainputs` and `dsa_configsys` scripts are located under the `/opt/teradata/datamover/support` directory.

1. From the DSC host, run `hostname -i` to verify that the resolved IP can be reached from the TPA nodes.
2. Run `build_dsainputs` on one of the following servers, depending on which DSA configuration you are using:

DSA Configuration	Server
Bundled DSA	Data Mover server
External DSA	External DSC server Copy the <code>build_dsainputs</code> script to the external DSC server and run from there. The script can be copied to and run from any directory.

3. Configure the source and target systems and restart DSMAIN using `dsa_configsys`. The following is an example of using `dsa_configsys` to configure TDPID `sdt05126` and restart DSMAIN:

```
# ./dsa_configsys -t sdt05126 -p dbc -h sdt05126 -u root
{"validationlist":null,"status":"CONFIG_TERADATA_SYSTEM_SUCCESSFUL","systemName":
:"sdt05126","tdpId":"sdt05126","valid":true,"links":[]}
```

sdt05126 successfully configured.

-

Preparing to stop/start DSMain.

There may be a delay of 30 seconds or more after each operation.

PLEASE ALLOW BOTH OPERATIONS TO COMPLETE.

If prompted .. please enter the password for user root on sdt05126:

```

Teradata

```

Release 15.10.03.05 Version 15.10.03.05
DSA DSMain Utility

This program is used to start or stop DSMain

** DSMain is not running **

```

Teradata

```

Release 15.10.03.05 Version 15.10.03.05
DSA DSMain Utility

This program is used to start or stop DSMain

DSMain started.

Restart of DSMain Complete.

Verifying that sdt05126 has registered with ActiveMQ .. please wait.

sdt05126 has been successfully configured and has registered with ActiveMQ.

To check if a system has been configured with DSA, use the --verify option with the -t and -p options.

The following example shows the dsa_configsys usage information that is displayed when no parameters are provided:

```

-----
dsa_configsys

```

```

Usage: dsa_configsys -t <TDPID> [-U <db_user>] -p <db_user password> [--verify]
[ -h <host> -u <linux userid> [-i <identity_file>] ] | -?

```

Description:

This script will configure a TD system for DSA (similar to the 'dsc config_systems' command). After the TD system is configured DSMain must be re-started on the system. This script will restart DSMain if the -h option is specified.

Run this script locally on the DSC server for which the TD systems are

being configured.

- t The TDPID of the system to configure. The TDPID must exactly match the definition from the Data Mover <source_tdpid> and <target_tdpid> job XML. For example: if the <source_tdpid> in the job XML is prodsys.labs.td.com then it must be specified as prodsys.labs.td.com for this option.
- U (Optional) Use this option to specify a user other than 'dbc'. This user must have adequate permission for DB 'sysbar' to replace a macro (TD version < 16.00) or insert a row in table sysbar.dsaconnectionstbl (TD version >= 16.00).
- p Password for user 'dbc' on the system specified above with the -t option. Optionally, the -U option can be added to specify a different user. Password must be enclosed by single quotes.
- verify (Optional) Use this option along with -t and -p to check whether a system has been configured in DSA. If the system is configured then no additional configuration is needed with dsa_configsys.
- h (Optional) If the user specifies the -h option followed by the TD system host (SMP) or one of the TD system hosts (MPP) then dsa_configsys will ssh into that host to restart DSMain (stop/start).
- u Required when using the -h option, specify a user that has sudo permissions to run TD cnstern commands on the host (e.g. root, ec2-user, azureuser).
- i (Optional) Specify an identity file that includes the private key.

Examples:

```
./dsa_configsys -t devsys -p dbc -h devsys -u root
Configure SMP system 'devsys' and restart DSMain.
```

```
./dsa_configsys -t prodsys.labs.td.com -p dbc -h prodsys1 -i
my_identityfile -u ec2-user
Configure MPP system 'prodsys' and restart DSMain. Note that
'prodsys1' resolves to the first node of the MPP system.
```

```
./dsa_configsys -t prodsys -p dbc --verify
Check if 'prodsys' has been configured in DSA.
```

-? Displays usage help.

System Aliases

Host aliases, or tdpids, for source and target systems are defined either in DNS (preferred) or in hosts file of all Data Mover-DSA components such as the servers and source and target TPA nodes.

If a system has been previously configured with DSA using a tdpid and you want to refer to that system through a different server alias in the job definition, you can add the server alias to the DNS or hosts file. There is no need to reconfigure the DSA system, provided that the previously configured tdpid remains working.

When configuring DSA components, be sure to refer to the server aliases instead of the IPs for the hosts involved. For example, use the server alias in `dsc.properties` and `clienthandler.properties`, as well as the ActiveMQ brokers, BAR NC master nodes, and when configuring DSA system tdpid.

Installing and Configuring BAR NC on TPA Nodes

1. Run `build_dsainputs` on one of the following servers, depending on which DSA configuration you are using:

DSA Configuration	Server
Bundled DSA	Data Mover server
External DSA	External DSC server Copy the <code>build_dsainputs</code> script to the external DSC server and run from there. The script can be copied to and run from any directory.

2. Create a `dsainputs` file for the TPA node using the `build_dsainputs` tool.
The `build_dsainputs` tool creates the `/tmp/dsainputs` files for SMP and MPP databases. It optionally transfers the files to the TPA nodes if the user with write access to the `/tmp` directory is specified with the `-u` option. An identity file that contains a private key can also be specified with the `-i` option.

The following example shows the output when no parameters are provided when running `build_dsainputs`:

```
build_dsainputs -d [-t] | -m <media_server_hostname>[,<media_server_hostname>...] | [-i <identity_file> -u
<userid>] | [-u <userid>] | -h
```

Parameters:

- `-d` Creates a `/tmp/dsainputs` file for the DSC server.
- `-m <ms_hostname>[,<ms_hostname>...]` Creates `/tmp/dsainputs` file(s) for a media server. Run this script one time for each Teradata system. Specify one host for an SMP System or a comma separated list for an MPP System. You may also specify a range of hostnames using `<ms_hostname>[##-##]` syntax. Creates a `/tmp/dsainputs.<media_server_hostname>` file for each host in the list. The `/tmp/dsainputs` files must be copied to their corresponding host before installing the ClientHandler rpm.

Examples:

- 1) `build_dsainputs -m devsys`
Creates `/tmp/dsainputs.devsys` for SMP system 'devsys'. Copy this file to the corresponding database host and rename to `/tmp/dsainputs` before installing the ClientHandler rpm.
- 2) `build_dsainputs -m kiwi1,kiwi2`
Creates `dsainputs` files for a 2-node MPP system with hosts `kiwi1` and `kiwi2`. The following files are created:

```
/tmp/dsainputs.kiwi1
/tmp/dsainputs.kiwi2
```

Copy these files to the corresponding database hosts (rename `/tmp/dsainputs`) before installing the ClientHandler rpm.

- 3) `build_dsainputs -m cucumber[1-3],onion1,tomato[12-13]`

```
/tmp/dsainputs.cucumber1
/tmp/dsainputs.cucumber2
/tmp/dsainputs.cucumber3
/tmp/dsainputs.onion1
/tmp/dsainputs.tomato12
/tmp/dsainputs.tomato13
```

Copy these files to the corresponding database hosts (rename to `/tmp/dsainputs`) before installing the ClientHandler rpm.

- `-i <identity_file>` (Optional) When used with the `-m` option this specifies the

scp	identify_file (that contains the private key) that will be used along with the userid (provided with the -u option) to the /tmp/dsainputs(s) files to their respective hosts. The -u option must be provided when the -i option is included.
-u <userid>	Example: build_dsainputs -m kiwi1,kiwi2 -i ./my_identity_file -u ec2-user
to	(Optional) When used with the -m option this specifies the userid that will be used to scp the /tmp/dsainputs(s) files
configured,	their respective host(s). Note that if you are using this without the -i option and password-less SSH is not
	you will be prompted for a password for each file transfer.
-t	Example: build_dsainputs -m kiwi1,kiwi2 -u root
	(Optional) Must be used with -d option to generate the DSA input file. If specified, Teradata-specific parameters are added to /tmp/dsainputs.
-h	If not specified, PostgreSQL parameters are added instead.
	Displays usage help.

The following example shows running build_dsainputs for a 4-node MPP system. The -u option has been provided to transfer the files to the individual TPA nodes:

```
# ./build_dsainputs -m kiwi1,kiwi2,kiwi3,kiwi4 -u root
Generating /tmp/dsainputs.kiwi1 - copy this file to /tmp/dsainputs on kiwi1 before
installing the ClientHandler rpm.
Generating /tmp/dsainputs.kiwi2 - copy this file to /tmp/dsainputs on kiwi2 before
installing the ClientHandler rpm.
Generating /tmp/dsainputs.kiwi3 - copy this file to /tmp/dsainputs on kiwi3 before
installing the ClientHandler rpm.
Generating /tmp/dsainputs.kiwi4 - copy this file to /tmp/dsainputs on kiwi4 before
installing the ClientHandler rpm.

Transferring /tmp/dsainputs file to kiwi1.
dsainputs.kiwi1
349      0.3KB/s   00:00                               100%

Transferring /tmp/dsainputs file to kiwi2.
dsainputs.kiwi2
348      0.3KB/s   00:00                               100%

Transferring /tmp/dsainputs file to kiwi3.
dsainputs.kiwi3
348      0.3KB/s   00:00                               100%

Transferring /tmp/dsainputs file to kiwi4.
dsainputs.kiwi4
```

Note:

- For an MPP system, nodes listed in the media server with the -m option must resolve to IPs that are reachable from each MPP TPA node.
- When you are running the tool on an SMP system, only a single file generation and transfer message appears.

3. Install the DSA ClientHandler RPM on the TPA nodes.

For more information, see *Teradata® Data Stream Utility Installation, Configuration, and Upgrade Guide*, B035-3153.

Note:

If you are installing the clienthandler on the command line instead of using PUT, use the `rpm -ivh ClientHandler*.rpm` command instead of the bundled script. If you use the bundled `clienthandler_install.sh` script on the command line, the `/tmp/dsainputs` property settings are overwritten and you must remove the ClientHandler using `rpm -e` and then start over with step 1 to create the `dsainputs` file for the TPA node.

4. Perform the following steps to update the operating system on the BAR NC TPA nodes:
 - a. Adjust the following operating system tunable parameters on the TPA nodes where BAR NC is running:

```
net.ipv4.tcp_max_syn_backlog to 16384 (default 2048)
net.core.somaxconn to 16384 (default 128)
```

- b. Run the following commands on the TPA nodes to incorporate changes immediately:

```
# echo 16384 > /proc/sys/net/core/somaxconn
# echo 16384 > /proc/sys/net/ipv4/tcp_max_syn_backlog
```

- c. Restart the ClientHandler on those TPA nodes:

```
# /etc/init.d/clienthandler restart
```

- d. Modify the `sysctl.conf` with the following values to make the values persistent at startup:

```
net.core.somaxconn=16384
net.ipv4.tcp_max_syn_backlog=16384
```

Note:

Teradata recommends incorporating these changes to avoid network communication issues between DSMain and BAR NC. See KB0016835 for more details.

Installing AXMS3 Module on TPA Nodes

The BAR NC requires the AXMS3 (Access Module for Amazon S3 Targets) module to interact with Amazon S3.

Make sure, the ClientHandler and the AXMS3 RPMs version are same. The minimum version required for both RPMs to use with Cloud Staging Copy Service are available in the `/var/opt/teradata/packages/DataMover` directory on new systems, or in the current `tdm-linux` bundle tarball.

To install the AXMS3:

1. Extract the `AXMS3*.tar.gz` file into a `/tmp` folder from the `/var/opt/teradata/packages/DataMover` directory.

2. Copy the AXMS3*.rpm file from the /tmp folder to each source and target node.
3. Install the AXMS3*.rpm on each node: `rpm -ivh AXMS3*.rpm`

Cloud Staging Area Setup

You need at least one Cloud Staging Area to create Data Mover jobs using Cloud Staging Copy Service. For creating a Cloud Staging Area you need your AWS account and S3 bucket information as following:

- AWS Account Id
- Secret Access Key

Following are the two options for creating a Cloud Staging Area:

- Provide AWS S3 account information and allow Data Mover through Cloud Staging Copy Service to define the target groups and map automatically.
- Provide the name of the target group that have already been defined in DSC and paired in a Target Group Map.

Using AWS S3 Credentials

To create the Cloud Staging Area:

1. Run the command `create_cloud_staging` to open `createCloudStaging.xml`.

```
?xml version='1.0' encoding='UTF-8'?>
<dmCreateCloudStaging xmlns="http://schemas.teradata.com/dataMover/v2009"
xsi:schemaLocation="http://schemas.teradata.com/unity/DataMover.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <name>csa1</name>
  <storage_type>S3</storage_type>
  <s3_properties>
    <access_key_id>ACCESSKEYID</access_key_id>
    <secret_access_key>SecretAccessKey</secret_access_key>
    <buckets_by_regions>
      <buckets_by_region>
        <region>us-west-2</region>
        <buckets>
          <bucket>
            <bucket_name>my-s3-bucket</bucket_name>
            <prefix_list>
              <prefix>
                <prefix_name>backup</prefix_name>
                <storage_devices>100</storage_devices>
              </prefix>
            </prefix_list>
          </bucket>
```

```

        </buckets>
      </buckets_by_region>
    </buckets_by_regions>
  </s3_properties>
  <source_target_pairs>
    <source_target_pair>
      <source_system>sourceSystem1</source_system>
      <target_system>targetSystem1</target_system>
    </source_target_pair>
  </source_target_pairs>
</dmCreateCloudStaging>

```

2. Provide your AWS account and S3 bucket information in the `createCloudStaging.xml` file and save.
3. Run the command: `datamove create_cloud_staging -f createCloudStaging.xml`.

Note:

If you have not specify the **secret_access_key** in the xml, the command line interface prompts you to enter the secret access key.

At the end of the script, the command prompts the completion message or reports any errors if occurred.

For further information on the Cloud Staging Area, refer to the *Teradata® Data Mover User Guide*, B035-4101

Using Configured Target Group Names

To create the cloud Staging Area:

1. Configure the AWS account with DSC.
 - a. Run the command `dsc config_aws -file aws_config.xml`. The `aws_config.xml` contains:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dscConfigAmzS3 xmlns="http://schemas.teradata.com/v2012/DSC">
  <config_aws_list>
    <account_name>aws-account-name</account_name>
    <buckets_by_region>
      <region>us-west-2</region>
      <buckets>
        <bucket_name>my-s3-bucket</bucket_name>
        <prefix_list>
          <prefix_name>backup</prefix_name>
          <storage_devices>100</storage_devices>
        </prefix_list>
      </buckets>
    </buckets_by_region>
  </config_aws_list>
</dscConfigAmzS3>

```

```

        </prefix_list>
    </buckets>
</buckets_by_region>
</config_aws_list>
</dscConfigAmzS3>

```

- b. Enter the AWS Access Id and Key separately as the script prompts.
2. Configure the target group with DSC for each node.
 - a. Run the command `dsc config_target_groups -type TARGET_S3 -file aws_source_tg.xml`. The `aws_source_tg.xml` contains:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dscConfigTargetGroupsAmzS3 xmlns="http://schemas.teradata.com/
v2012/DSC">
    <target_group_name>source-tg-name</target_group_name>
    <is_enabled>true</is_enabled>
    <account_name>aws-account-name</account_name>
    <region>us-west-2</region>
    <targetMediaBuckets>
        <bar_media_server>sourceSystem1_ms</bar_media_server>
        <buckets>
            <bucket_name>my-s3-bucket</bucket_name>
            <prefix_list>
                <prefix_name>backup</prefix_name>
                <storage_devices>100</storage_devices>
            </prefix_list>
        </buckets>
    </targetMediaBuckets>
</dscConfigTargetGroupsAmzS3>

```

- b. Enter the name of the media server for the source or the target in **bar_media_server**.
- c. Enter the name of the AWS account in **account_name**.
3. Run the command `dsc config_target_group_map -f aws_tg_map.xml` to configure the target group map with DSC. The `aws_tg_map.xml` contains:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dscConfigTargetGroupsAmzS3 xmlns="http://schemas.teradata.com/v2012/DSC">
    <target_group_name>source-tg-name</target_group_name>
    <is_enabled>true</is_enabled>
    <account_name>aws-account-name</account_name>
    <region>us-west-2</region>
    <targetMediaBuckets>
        <bar_media_server>sourceSystem1_ms</bar_media_server>
        <buckets>

```

```

        <bucket_name>my-s3-bucket</bucket_name>
        <prefix_list>
            <prefix_name>backup</prefix_name>
            <storage_devices>100</storage_devices>
        </prefix_list>
    </buckets>
</targetMediaBuckets>
</dscConfigTargetGroupsAmzS3>

```

- Specify the target group names in the createCloudStaging.xml. The createCloudStaging.xml contains:

```

<?xml version='1.0' encoding='UTF-8'?>
<dmCreateCloudStaging xmlns="http://schemas.teradata.com/dataMover/v2009"
xsi:schemaLocation="http://schemas.teradata.com/unity/DataMover.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <name>csa1</name>
    <storage_type>S3</storage_type>
    <source_target_pairs>
        <source_target_pair>
            <source_system>sourceSystem1</source_system>
            <source_system_target_group>source-tg-
name</source_system_target_group>
            <target_system>targetSystem1</target_system>
            <target_system_target_group>target-tg-
name</target_system_target_group>
        </source_target_pair>
    </source_target_pairs>
</dmCreateCloudStaging>

```

- Run the command `datamove create_cloud_staging -f createCloudStaging.xml` to create a cloud staging area.

For further information on the Cloud Staging Area, refer to the *Teradata® Data Mover User Guide*, B035-4101

DSA Network Configuration

DSA requires you to define which network paths to use when transferring data between the database and the DSA media servers/BAR NC. This is achieved by either configuring a network fabric or a logical netmask. Teradata recommends using the network fabric method in most cases for the following reasons:

- Network fabrics define a specific path which is better for supportability and performance
- Logical netmasks require removing unwanted IPs for all nodes

- Network fabrics make it easier to support the unfolding and folding of the database in public clouds

Network Fabric for Systems

A network fabric defines the data path between one or more nodes in a Teradata system and DSA media servers using the most efficient interface when transferring data. In a Data Mover-DSA configuration, media servers are installed on the target system nodes.

Network fabrics are defined by using one of the following methods:

- The `dsa_configfabric` script provided in the Data Mover installation package
 - You can also find the script under the `/opt/teradata/datamover/support/dsa_configfabric` directory.
- The `config_fabric` command in the DSC command line
- The DSA portlet

The DSA portlet version must match the DSC version. See *Teradata® Data Stream Utility Installation, Configuration, and Upgrade Guide*, B035-3153 for more information.

For example, a configuration where the media server on SystemA is connected in the following way:

- to System B using both a high-speed BYNET network (10.16.xx.xx) and a LAN network (10.25.xx.xx)
- to System C using only a LAN network (10.25.xx.xx)

In this example, defining network fabrics for System B and System C makes sure that traffic from System B to System A uses the high-speed BYNET, while System C to System A uses the LAN network. The difference, when compared to a logical netmask configuration, is that while the netmask allows traffic from the media server to both System B and System C, it cannot limit the System B traffic to use only the high-speed BYNET.

Below is a sample fabric XML for System B, specifying the media server IP addresses to be used on System A. This is using a configuration where both system A and B are 2-node MPP systems. A similar fabric XML is needed to configure a fabric for System C. For more information, see *Teradata® DSA User Guide*, B035-3150.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dscConfigFabrics xmlns="http://schemas.teradata.com/v2012/DSC">
  <fabric_name>systemBFabricConfig</fabric_name>
  <system_name>systemB</system_name>
  <is_enabled>true</is_enabled>
  <tpa_to_ms>
    <node_name>systemB-01</node_name>
    <ms_info>
      <media_server>systemAcop1_ms</media_server>
      <ip_address>10.16.108.150</ip_address>
```

```

    </ms_info>
  </tpa_to_ms>
  <tpa_to_ms>
    <node_name>systemB-01</node_name>
    <ms_info>
      <media_server>systemAcop2_ms</media_server>
      <ip_address>10.16.108.63</ip_address>
    </ms_info>
  </tpa_to_ms>
  <tpa_to_ms>
    <node_name>systemB-02</node_name>
    <ms_info>
      <media_server>systemAcop1_ms</media_server>
      <ip_address>10.16.108.150</ip_address>
    </ms_info>
  </tpa_to_ms>
  <tpa_to_ms>
    <node_name>systemB-02</node_name>
    <ms_info>
      <media_server>systemAcop2_ms</media_server>
      <ip_address>10.16.108.63</ip_address>
    </ms_info>
  </tpa_to_ms>
</dscConfigFabrics>

```

Configuring a Network Fabric

Data Mover provides the `dsa_configfabric` script as part of the installation package to make the fabric configuration process faster and easier. The script generates two XML files that contain the source and target system fabric definitions. The script uses DSC commands to extract information about the source and target media servers and nodes. It then generates the following two fabric files:

- One file defines the connection from source nodes to target media servers and source nodes to source media servers
 - The other file defines the connection from the target nodes to source media servers and target nodes to target media servers
1. Before running the configuration script, make sure that all interfaces that will be part of the fabric are available in the DSC media server configuration.
 2. [Optional] Use one of the following two methods if interfaces are missing from the DSC configuration:

Method	Description
Manually export, add interfaces, and reconfigure each media server	a. Export the media server configuration: <code>dsc export_config -t MEDIA_SERVER -n <i>media_server_name</i> -f <i>media_server_XML_config_filename</i></code> b. Edit the <i>media_server_XML_config_filename</i> and add the missing interfaces. c. Save the changes: <code>dsc config_media_servers -f <i>media_server_XML_config_filename</i></code>
Reset all the media server settings registered with DSC	Run the following command: <code>dsc config_media_servers -f <i>media_server_XML_config_filename</i></code> Note: Using the hardware upgrade script resets the settings for all the DSCs using that media server. Use with caution.

The media server configuration should look similar to the example after the hardware resets.

```

Media Server Name   Port    Pool Shared Pipes   IP Address(es)
NetMask(s)
-----
-----
server1_ms          15401   100                10.0.***.***
255.255.254.0                                10.25.***.***
255.255.254.0                                10.1.***.***
255.255.254.0                                10.2.***.***
255.255.254.0
-----
-----
server2_ms          15401   100                10.1.***.***
255.255.254.0                                10.2.***.***
255.255.254.0                                10.0.***.***
255.255.254.0                                10.25.***.*** 255.255.254.0

```

- Use the following DSC commands to confirm the media servers and system nodes are recognized by DSC:

- `dsc list_components -t system`
- `dsc list_components -t media_server`

- `dsc list_components -t target_group`

4. Run the script to generate the fabric for the source and target systems.

Use the following help command output for the list of available script parameters:

```
HELP: dsa_configfabric -s [source_system] -t [target_system]
-s | --source [source_system]      Source system name
-t | --target [target_system]      Target system name
-f | --file [accept_list_file_path] The file contains one or combination of
the following two entries:
    1. Interface IP address, Node name, and Media Server
name defined by DSC separated by tabs
    2. Interface IP address defined in DSC
If the line entry is applicable to the questionnaire, an
interface for the connection is selected automatically based on the priorities
specified earlier.
-h | --help                        Help
```

Note:

This script requires Python 2.6.9 and must run on the machine where DSC is installed.

5. Add the generated DSA fabric configuration files to the DSC configuration:

```
dsc config_fabrics -f ./output/xml_generated_file_name.xml
```

Configuring TPA Nodes without Remote Connectivity

When one or more TPA nodes on a source or target system does not have remote connectivity, the DSA fabric must be configured for both DSA systems specifying the connectivity between system nodes and media server node IPs.

- Configure the DSA network fabrics for the source and target systems to include the following:
 - Define the source system fabric to include only the paths where source TPA nodes have remote connectivity to the target media server node IPs.
 - Define the target system fabric to include only the paths where target TPA nodes have remote connectivity to its own media server node IPs.
 - If media servers are also installed on source system nodes, do the following:

System Fabric	Description
Source	Add paths where the source TPA nodes have remote connectivity to its own media server node IPs.
Target	Add paths where the target TPA nodes have remote connectivity to the source media server node IPs.

- Change the following value in `dsc.properties` to turn fabric validation off to avoid job failure:


```
# Fabric validation all nodes need a path to a media server
#Possible values On/Off
Validation.fabric=off
```

3. Restart the DSC.

Configuring Data across Multiple Clouds

When copying data between systems across multiple cloud platforms, DSA fabrics must be configured for both systems.

Note:

This feature does not work unless you have an elastic IP service enabled.

1. Verify both private and public IPs have been defined for each media server.
2. Open required ports for Data Mover-DSA configuration across cloud platforms, this includes ActiveMQ, database, and media server ports.
3. Create server aliases for all components involved, such as the Data Mover and DSA servers and the source and target systems.
4. Define the server alias to IP mapping in hosts file.
Map public IPs to servers in the external cloud and private IPs to servers in the internal cloud.
5. Define fabrics for the source and target systems using the following requirements:
 - a. Define the source system fabric to specify paths from the source TPA nodes to the public IPs of the target media server nodes.
 - b. Define the target system fabric to specify paths from the target TPA nodes to the private IPs of its own media server nodes.
 - c. If media servers are also installed on source system nodes, do the following:

System Fabric	Description
Source	Add paths from the source TPA nodes to the private IPs of its own media server nodes.
Target	Add paths from the target TPA nodes to the public IPs of the source media server nodes.

Node Failure Recovery (NFR) Considerations for Cloud

Node Failure Recovery (NFR) is a cloud feature wherein Teradata SQL-E nodes are automatically replaced when a node fails. When NFR occurs, only static IP addresses associated with that node remains valid while non-static IP addresses are replaced. If a network fabric is defined using non-static IP addresses, that network fabric becomes invalid after NFR occurs.

To avoid this scenario, choose static IP addresses when defining a network fabric for a cloud-based SQL-E system. In AWS, these are referred to as elastic IP addresses and in Azure, these are referred to as static public IP addresses. On a SQL-E node, these are generally associated with the `eth0:0` to `eth0:3` network interfaces. Consult with a cloud expert when defining a network fabric to make sure that the IP addresses chosen are static IP addresses that are considered valid in the case of NFR.

Network Fabric Configurations

Use the following guidelines when determining the number of connections to define in a network fabric:

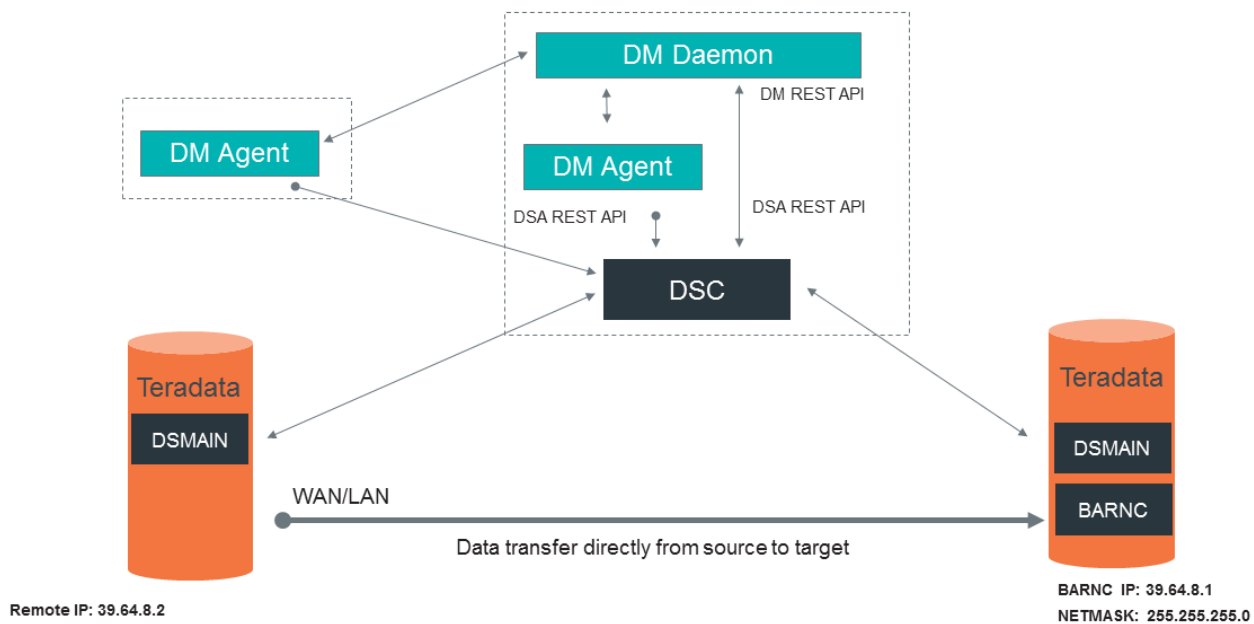
Scenario	Number of Connections
Copying data between a source and target system that have same number of nodes	Define 1:1 connections
Copying data from a source that has fewer nodes than the target system	Define $N < M$ Define N connections where N is the number of source nodes and M is the number of target nodes. If some N:M paths are not defined, define N connections.
Copying data from a source that has more nodes than the target system	Define $N > M$ Define N connections where N is the number of source nodes and M is the number of target nodes. Use a round-robin of target nodes to cover each source node.

BAR NC Logical Netmask

The logical netmask defines the communication between the BAR NC and the remote Teradata systems by determining which network interface the remote database uses for data transfer to BAR NC.

Note:

If you are using an external DSC and the source or target Teradata system is configured to use a network fabric, the Data Mover media server (clienthandler) information must be configured to the fabric rather than the logical netmask. For more information, see [Network Fabric for Systems](#).



Teradata recommends that you restrict the netmask to limit traffic to high-speed networks only. For example, if the source and target systems are in the same 39.64.8.x BYNET network, a netmask of 255.255.255.0 will make sure there is communication from the remote IP to the BAR NC over the high bandwidth BYNET network. While a netmask of 0.0.0.0 does work, traffic over any of the network interfaces is allowed, such as going through a low-bandwidth 10.25.22.x LAN network, which may not be desirable.

Configuring the Logical Netmask for ClientHandler

BAR NC supports the configuration of multiple networks and logical netmasks. The following is an example of a BAR NC configuration with multiple network interfaces:

```
<dscConfigMediaServers xmlns="http://schemas.teradata.com/v2012/DSC">
  <media_server_name>sdt05126_ms</media_server_name>
  <port>15401</port>
  <ip_info>
    <ip_address>39.64.8.1</ip_address>
    <netmask>255.255.255.0</netmask>
  </ip_info>
  <ip_info>
    <ip_address>10.25.188.87</ip_address>
    <netmask>255.255.0.0</netmask>
  </ip_info>
</dscConfigMediaServers>
```

Note:

You can use the DSA portlet to configure netmasks instead of the DSC command line and XML files, if one is available in your environment. The DSA portlet version must match the DSC version.

The steps here are an example of a BAR NC configuration with single network interfaces.

1. List the media server:

```
# dsc list_components -t MEDIA_SERVER
Data Stream Controller Command Line 16.20.00.00
Command parameters:
-type : MEDIA_SERVER
Connected to DSC version 16.20.00.00
Listing Components...
Media Server Name Port Pool Shared Pipes IP Address(es) NetMask(s)
-----
---
sdt05126_ms 15401 100 10.25.188.87 255.255.254.0
-----
---
```

2. For each media server export the configuration to an XML file:

```
# dsc export_config -t MEDIA_SERVER -n sdt05126_ms -f sdt05126_ms.xml
Data Stream Controller Command Line 16.20.00.00
Command parameters:
-type : MEDIA_SERVER
-name : sdt05126_ms
-file : sdt05126_ms.xml
Connected to DSC version 16.20.00.00
Exporting Configurations...
Export config successful!
```

3. Edit the exported XML file and set the netmask to the desired value:

```
# cat sdt05126_ms.xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dscConfigMediaServers xmlns="http://schemas.teradata.com/v2012/DSC">
  <media_server_name>sdt05126_ms</media_server_name>
  <port>15401</port>
  <pool_shared_pipes>100</pool_shared_pipes>
  <ip_info>
    <ip_address>10.25.188.87</ip_address>
    <netmask>255.255.254.0</netmask>
  </ip_info>
</dscConfigMediaServers>
```

Note:

- The example shows the values before editing the netmask to another value.
- Leave only the IP addresses intended for data transfer in the exported XML file. Remove all other IP addresses, including their surrounding `<ip_info>` `</ip_info>` xml tags.

4. Update the configuration with the new values with `dsc config_media_servers`:

```
# dsc config_media_servers -f sdt05126_ms.xml
Data Stream Controller Command Line 16.20.00.00
Command parameters:
-file : sdt05126_ms.xml
Connected to DSC version 16.20.00.00
Saving Media Server...
Media Server config succeeded for "sdt05126_ms"
```

5. List the media servers again to confirm that the netmask value has been changed:

```
# dsc list_components -t MEDIA_SERVER
Data Stream Controller Command Line 16.20.00.00
Command parameters:
-type : MEDIA_SERVER
Connected to DSC version 16.20.00.00
Media Server Name Port Pool Shared Pipes IP Address(es) NetMask(s)
Listing Components...
-----
---
sdt05126_ms 15401 100 10.25.188.87 255.255.0.0
-----
---
```

Enabling TLS 1.2 Data Path Encryption for DSC on the Data Mover Server

Note:

This is an optional configuration

To enable TLS 1.2 encryption in DSA data path for DM DSA jobs using DM self signed certificate, run the `dsa_tlscert.py` script available in packages directory under the `dmdsa_TLSconfig` directory.

Note:

- Both the source and the target systems must support TLSv1.2 encryption. To verify, run following command on the system: "openssl ciphers -v | grep TLS"
- Script needs to connect from the DM/DSC Daemon server to port 22 on source/target systems to copy files and run commands. Make sure ports are enabled.
- If port 22 does not open due to security reasons, please reach out to customer support for assistance with the manual configuration of DSA TLS 1.2 encryption enablement.

```

HELP: dsa_tlscert.py --source <source_TDPID> --source_username
<source_system_username> --source_password <source_system_password>[/ --
source_identity_file <identity_file>] --target <target_TDPID> --target_username
<target_system_username> --target_password <target_system_password>[/ --
target_identity_file <identity_file>] ] ]
--help                | --
help                  | Displays Help
--source              | --TDPID/name of source
system                | Source system name
--source_username     | --source sytem user that has sudo
permission            | Ex: root, ec-user, azureuser, gcpuser
--source_password     | --Source sudo user password or
--source_identity_file | --identity file that includes the private key **
--target              | --TDPID/name of Target
system                | Target system name
--target_usrname      | --Target sytem user that has sudo
permission            | Ex: root, ec-user, azureuser, gcpuser
--target_password     | --Target sudo user password or
--target_identity_file | --identity file that includes the private key **

```

Example command: `python3 dsa_tlscert.py --source sdt35178 --source_username root --source_password datamover --target sdt35179 --target_username root --target_password datamover`

This script enables TLS 1.2 Encryption between source sss and target ttt using self-signed certificates for DM DSA jobs.

Note:

This script does not support an existing BAR DSA TLS 1.2 certificate on a shared clienthandler.

1. Generate a new pair of self-signed certificates for source sss and target ttt.
 - This generates the TLS 1.2 certificates for DSMAIN and clienthandler

- Copies `clienthandler.properties` to `clienthandler.properties.dm_dsaTLS`, with added TLS properties
- Runs `enableTLS_dsc.sh` script, which copies `dsc.properties` to `dsc.properties.tls`, with added TLS properties

Note:

This option overwrites the certificates on `sss` and `ttt` if it already exist.

If one system has already been TLS 1.2 encrypted with another third system, choose option 2 or 3.

2. Generate a new self-signed certificate for source `sss` and use existing DM generated self-signed certificate from target `ttt`.
3. Use existing DM generated self-signed certificate from source `sss` and generate a new self-signed certificate for target `ttt`.

To finish TLS 1.2 encryption, perform the following manual steps when `DSMAIN` and `clienthandler` can be restarted:

- For the newly encrypted DSA system with `clienthandler` installed, copy `clienthandler.properties.dm_dsaTLS` to `clienthandler.properties` and restart `clienthandler` on all nodes
- If DSC is not previously TLS enabled, copy `dsc.properties.tls` to `dsc.properties` and restart DSC
- Reconfigure the newly encrypted DSA system using `dsa_configsys` or `dsc commandline`

Note:

To disable TLS 1.2 encryption:

1. Edit `dsc.properties` to set `tls.datapath.enabled=false`
 2. Restart DSC
 3. Re-configure source system and restart `dsmain`, using `dsa_configsys` or `dsc commandline`
 4. Re-configure target system and restart `dsmain`, using `dsa_configsys` or `dsc commandline`.
-

Deploying Data Mover on VMware

About Data Mover on VMware

Data Mover is available on a VMware template to run in a virtual environment on third-party hardware. The Data Mover VMware template hosts the following:

- Data Mover software
- Postgres repository
- SUSE Linux Enterprise Server (SLES) operating system

This information is intended for those experienced in vSphere, vCenter, ESXi terminology, and who have an understanding of standard switches, datastores, VM templates, .ova, .ovf, and other database components.

Requirements for Data Mover on VMware

Component	Minimum Requirements (unless noted)
vCPU	4
Memory	32GB
Datastores for database storage	One datastore
Datastore formatting	VMFS 5 file system
Datastore space for template deployment	300GB
RAID	RAID-1 LUNs recommended.
Server	Dell PowerEdge R720xd/R730xd or an HP Gen9 or similar system. Hyper-threading (HT) must be enabled.
Network	10Gb recommended
VM deployment size	300GB
VMware ESXi	ESXi 5.5 release 2 and later
VMware licensing	A full VMware vSphere 6.0 license or VMware vSphere Essentials 6.0 license for at least one physical CPU (unlimited cores per CPU)
VMware vCenter	vCenter 5.5 release 2 and later
VMware vSphere switches	One vSphere standard vSwitch for public network.
Operating System	SUSE Linux Enterprise Server (SLES)

Downloading Data Mover for VMware

1. Go to <https://support.teradata.com>.
2. Log in.
3. Select **Update Your Software**.
4. From the Navigation panel, select **Database and Applications**.
5. Select the **Teradata Applications** tab and do the following:
 - a. From **Sub Category**, select **Unity Data Mover**.
 - b. From **Release** select the release version to download.
 - c. As applicable, select the appropriate **Platform**, **Site Id**, and **Change Control**.
 - d. Select **Submit**.
6. Select the version of the .ova file to download, such as the following example package name:
Vantage_on_VMware_DM_17.00.00.00
7. At the top of the page, in **Teradata FTP/SFTP**, enter the FTP directory name.
8. Select the **Download** or **Download All** button.
The .ova file, which contains the template, is large (5GB) and may take time to download.
9. In the **Download** window, follow the instruction for accessing the software from the FTP/SFTP location.

Deploying Data Mover on VMware

1. Connect to the vCenter server to deploy Data Mover using the vSphere Client.
2. From the **Home** tab, go to **Inventories > Hosts and Clusters**.
3. In the **Navigator** pane, select a host and **Deploy OVF Template**.
4. Choose **Select an OVF template** and then:
 - a. Select **Local file**.
 - b. Select **Choose Files** and browse to the .ova file that you downloaded.
 - c. Click **Next**.
5. Add a public network to the instance after deploying the .ova file.
Wait for the deployment to complete.
6. In the **Navigator** pane, right-click the VM and select **Power > Power On**.
7. Once the VM instance is running, log on to the network using SSH with the following credentials:
linux login: root
Password: tdc
The IP address is displayed on the **Summary** screen of the vSphere client.
8. After login, configure the network:
 - a. Gather the following information to run the network configuration script:
 - VM name (enter a hostname for this VM)
 - Public IP address (must be static)
 - Gateway IP address

- Network domain
- Primary DNS IP address
- (Optional) Secondary DNS IP address
- (Optional) Private IP address – only enter if a separate private network adapter is attached to the VM and you want to configure it.
- Primary NTP server
- (Optional) Secondary NTP server

- b. Run the network configuration script:

```
# /var/opt/teradata/packages/DataMover/config-network-dm
```

Allow several minutes for the script to complete configuration – before and after it restarts the VM.

You can be run again at any time.

9. Run `# /opt/teradata/datamover/daemon/xx.xx/dm-control.sh status` to verify all services are running:

```
Checking for Teradata ActiveMQ:           running
Checking for Teradata DataMover dmdaemon:  running
dmdaemon is RUNNING
Checking for Teradata DataMover dmagent:    running
dmagent is RUNNING
Checking for TMSMonitor:                   running
Checking for Teradata BAR dsc:              running
dsc is running

Checking for DSARest web service:          running
The DSARest web service is running
```

10. Configure the bundled DSA using the following steps in the order shown:

- [Install and configure BAR NC on the TPA nodes.](#)
- [Configure the logical netmask for ClientHandler.](#)
- [Configure the source and target systems.](#)

11. Follow the steps in [Configuring ActiveMQ on Remote Agents](#) if this is the first time deploying or upgrading a standalone agent using this script.

Once a standalone remote agent has been set up manually, the settings are preserved across upgrades.

Administrative Tasks

Data Mover Components Script

Data Mover includes a single script that enables you to check status of, start, restart, or stop each Data Mover Component. The script name is `dm-control.sh`, and is installed by the Data Mover daemon package on the Data Mover Multi-Purpose Server in the directory: `/opt/teradata/datamover/daemon/17.xx.xx.xx`.

Note:

Do not use the `dm-control.sh` script when Data Mover is configured with high availability. To avoid interference with automatic failover, run `dmcluster status` from `/opt/teradata/client/nn.nn/failover` to check component status. If failover is configured, running `dmcluster setmaster` restarts services.

Note:

Only root users can run the `dm-control.sh` script or any of the individual Data Mover component scripts, including `tdactivemq`. Use the `dm-control.sh` to start/stop Data Mover services if Teradata is not recommending any other method.

The script includes the following commands:

Script Command	Description
<code>dm-control.sh status</code>	Displays status of Data Mover daemon, Data Mover agent, Data Mover sync, tmsmonitor, Teradata Database, Data Stream Controller, and Teradata ActiveMQ.
<code>dm-control.sh start</code>	Starts all Data Mover components on the local Data Mover Multi-Purpose Server. This includes Data Mover daemon, Data Mover agent, Teradata ActiveMQ, Data Stream Controller, and tmsmonitor. This does not include the Data Mover sync service or the failover monitoring service.
<code>dm-control.sh stop</code>	Stops all Data Mover components on the local Data Mover Multi-Purpose Server. This includes Data Mover daemon, Data Mover agent, Teradata ActiveMQ, Data Stream Controller, and tmsmonitor. This does not include the Data Mover sync service or the failover monitoring service.
<code>dm-control.sh restart</code>	Stops and starts all Data Mover components on the local Data Mover Multi-Purpose Server. This includes Data Mover daemon, Data Mover agent, Teradata ActiveMQ, Data Stream Controller, and tmsmonitor. This does not include the Data Mover sync service or the failover monitoring service.

Changing DBC, DATAMOVER, and DSA Passwords on the Data Mover Server

The script `changepassword.sh` allows the root user to change the DBC, DATAMOVER, BAR, and BARBACKUP passwords on the Data Mover server using Data Mover 17.00 or earlier.

Note:

The passwords for the bundled DSA in Data Mover are updated using the `changepassword.sh` script, do not use the script that is provided with DSA.

The new password specified for DATAMOVER using the script is applied to the users BAR and BACKUP. When started, the script logs into the internal TD repository and changes the repository passwords. The script is part of the `tdm-linux` bundle and its location is determined by where the bundle is located. For example:

```
/var/opt/teradata/packages/DataMover/16.20.12.00/changepassword.sh
```

Note:

If you previously used `dm.job.production.password` in the `daemon.properties` file to change the DATAMOVER password, remove it from `daemon.properties` before running the `changepassword.sh` script.

Note:

Teradata recommends running the script before installing or upgrading Data Mover. If you run at the script after Data Mover has been installed or upgraded, you must restart the daemon to make sure Data Mover runs properly.

1. Do one of the following:

Script Option	Description
Interactive	<ol style="list-style-type: none"> a. Run the script without arguments: # <code>changepassword.sh</code> b. When prompted, enter the old and new passwords.
Non-Interactive	<ol style="list-style-type: none"> a. Run the script with the arguments <code>-o</code>, <code>-p</code>, <code>-m</code>, and <code>-d</code>. For example: # <code>changepassword.sh -o old dbc password -p new dbc password -m old datamover password -d new datamover password</code>, where: <ul style="list-style-type: none"> • <code>old dbc password</code> is the existing password for the DBC user • <code>new dbc password</code> is the new password for the DBC user • <code>old datamover password</code> is the existing password for the DATAMOVER user • <code>new datamover password</code> is the new password for the DATAMOVER user

2. If you ran the `changepassword.sh` script after Data Mover was installed or upgraded, restart the daemon:
 - a. `# /etc/init.d/dmdaemon stop`
 - b. `# /etc/init.d/dmdaemon start`
3. Run the following commands to verify the Data Mover components are ready to use:
 - `datamove list_jobs`
 - `datamove list_agents`

Script Example

```
location:/var/opt/teradata/packages/DataMover162000 # ./changepassword.sh
Do you want to change the DBC password of the Teradata internal repository?
(yes/no/y/n)?
yes
-----
                Change DBC Default Password
-----
Old Password:
Retype Old Password:
New Password:
Retype New Password:
Do you want to change the DATAMOVER password of the Teradata internal
repository?(yes/no/y/n)?
yes
-----
                Change DATAMOVER/BAR/BARBACKUP Default Password
-----
Old Password:
Retype Old Password:
New Password:
Retype New Password:
*****
DBC default Password changed successfully

*****
*****
DATAMOVER default Password changed successfully
*****
*****
DBC and DATAMOVER passwords are encrypted and stored in /etc/opt/teradata/
datamover/password.txt file
*****
Please restart the Daemon using the following commands
```

```

/etc/init.d/dmdaemon stop
/etc/init.d/dmdaemon start
*****
*****
BAR default Password changed successfully
*****
*****
BARBACKUP default Password changed successfully
*****
*****
Please wait .. updating DSA properties files
*****
DSA properties files updated successfully
*****
The DSC and DSARest web service have been restarted
*****

```

Changing POSTGRES, DATAMOVER and DSA Passwords on the Data Mover Server

The script `change_pg_password.sh` allows the root user to change the POSTGRES, DATAMOVER, BAR, and CS2 passwords on the Data Mover server.

Note:

The passwords for the bundled DSA in Data Mover are updated using the `change_pg_password.sh` script, do not use the script that is provided with DSA.

The new password specified for DATAMOVER using the script is applied to the user BAR. When started, the script logs into the internal POSTGRES repository and changes the repository passwords. The script is part of the `tdm-linux` bundle and its location is determined by where the bundle is located on your server. For example:

```
/var/opt/teradata/packages/DataMover/17.12.00.00/change_pg_password.sh
```

Note:

If you previously used `dm.job.production.password` in the `daemon.properties` file to change the DATAMOVER password, remove the property from `daemon.properties` before running the `changepassword.sh` script.

Note:

Teradata recommends running the script before installing or upgrading Data Mover. Running the scripts after Data Mover has been installed or upgraded, requires restarting the daemon to make sure Data Mover runs properly.

The following special characters are not allowed in passwords:

- Single quote (')
- Double quotes (")
- Blank spaces

1. Do one of the following:

Script Option	Description
Interactive	a. Run the script without arguments: <code>change_pg_password.sh</code> b. When prompted, enter the old and new passwords.
Non-Interactive	a. Run the script with the arguments -o, -p, -m, and -d. For example: <code># change_pg_password.sh -o Current password for user 'postgres' -p New password for user 'postgres' -m Current password for user 'datamover', 'bar', & 'cs2' -d New password for user 'datamover', 'bar', & 'cs2', where:</code> <ul style="list-style-type: none"> • Current password for user 'postgres' is the existing password for the POSTGRES user • New password for user 'postgres' is the new password for the POSTGRES user • Current password for user 'datamover', 'bar', & 'cs2' is the existing password for the DATAMOVER, BAR, and CS2 user • New password for user 'datamover', 'bar', & 'cs2' is the new password for the DATAMOVER, BAR, and CS2 user

2. If you ran the `change_pg_password.sh` script after Data Mover was installed or upgraded, restart the daemon:

- `# /etc/init.d/dmdaemon stop`
- `# /etc/init.d/dmdaemon start`
- `# /etc/init.d/dmcs2 stop`
- `# /etc/init.d/dmcs2 start`

3. Use the following commands to verify the Data Mover components are ready to use:

- `datamove list_jobs`
- `datamove list_agents`

Script Example

```
location:/var/opt/teradata/packages/DataMover/17.12.00.00 # sh
change_pg_password.sh
Do you want to change password for user 'postgres' of the Postgres internal
repository?(yes/no/y/n)?
y
```

```

-----
                Change Postgres Password
Use double quotes when entering passwords containing special characters
-----
Old Password:
Retype Old Password:
New Password:
Retype New Password:
Do you want to change password for user 'DATAMOVER', 'BAR', and "CS2" of the
Postgres internal repository?(yes/no/y/n)?
y
-----
                Change DATAMOVER, BAR, and CS2 Password
Use double quotes when entering passwords containing special characters
-----
Old Password:
Retype Old Password:
New Password:
Retype New Password:
*****
*****
Postgres Password changed successfully
*****
*****
*****
*****
DATAMOVER, BAR, and CS2 Password changed successfully
*****
*****
DATAMOVER and BAR: Please restart the Daemon using the following commands
/etc/init.d/dmdaemon stop
/etc/init.d/dmdaemon start
*****
*****
CS2: Please restart the Daemon using the following commands
/etc/init.d/dmcs2 stop
/etc/init.d/dmcs2 start
*****
*****
Please wait .. updating DSA properties files
*****
*****
DSA properties files updated successfully
*****

```



```
*****
```

```
The DSC and DSARest web service have been restarted
```

```
*****
```

```
*****
```

Creating a Diagnostic Bundle for Support

For Data Mover situations such as job failure, job hanging, or other issues that require an incident report, Teradata includes interactive command-line scripts for collecting necessary job and system information. The resulting diagnostic bundle enables Teradata Customer Support to provide optimum analysis and resolution. Customer support is available around-the-clock, seven days a week through the Global Technical Support Center (GSC). To learn more, go to <https://support.teradata.com>.

The `dmagentsupport.sh` file collects the following information from a server running only the Data Mover agent:

- Data Mover log files from the agent server
- Recent temp and task directories

The `dmagentsupport.sh` script creates a `data-mover-agent-support` output file, which contains the following information:

- Data Mover `agent.properties` files
- List of files from the DataMover components installation directory
- OS, kernel, CPU, memory, and disk space information
- Data Mover and TTU packages rpm information

After the script collects the data, a bundle named `DataMover-$currentdate-$hostname-1.zip` is created in `/var/opt/teradata/datamover/support/incidentnumber`.

If the bundle size is larger than 49 MB, additional `.zip` files are created as follows:

- `DataMover-$currentdate-$hostname-2.zip`
- `DataMover-$currentdate-$hostname-3.zip`

1. Create a support incident including the following settings:

Option	Setting
Product Area	System Management Utilities
Problem Type	Teradata® Data Mover

2. Record the incident number and leave the incident open to attach the diagnostic bundle.

Note:

The interactive script prompts you to enter the incident number and other information related to the issue.

- As the root user, locate the scripts at `/opt/teradata/datamover/support/` for every Data Mover server in your environment, and do the following:

Server Type	Description
Data Mover Server	Run <code>dm-support.sh</code> to create a diagnostic bundle.
Server Running Only Data Mover Agent	Run <code>dm-agent-support.sh</code> to create a diagnostic bundle.

Be sure to include relevant problem descriptions for troubleshooting as prompted.

The `dm-support.sh` script collects the following information from the Data Mover log files on the Data Mover multi-purpose server:

- ActiveMQ queue information
- Recent temp and task directories
- DSA information, including:
 - DSC and DSA command line utility logs
 - Installation logs
 - Property files
 - RPM information

The script creates three output files:

Output File	Contents
<code>datamover-job-status</code>	<ul style="list-style-type: none"> Data Mover health information Data Mover and TTU packages rpm information List of total and failed Data Mover jobs List of job steps for failed jobs
<code>datamover-properties</code>	All data Mover properties files, including the following: <ul style="list-style-type: none"> List of files from the Data Mover components installation directory <code>ps aux</code> command output
<code>datamover-server-details</code>	OS, kernel, CPU, memory, and disk space information.

- Update the incident, browse to the resulting `.zip` files, attach the resulting files to the incident, and submit them.
- Contact Teradata Customer Support when the diagnostic bundle is ready for review, and include your incident number for reference.
- [Optional] If you do not want to keep the `.zip` files, delete them from the `/var/opt/teradata/datamover/support/incidentnumber` directory on the Data Mover server.

Note:

To capture jstack and jmap output for tdactivemq, daemon, and agent processes, add the option `-runjvmtools` as in the following examples:

```
dmsupport.sh -runjvmtools
```

```
dmagentsupport.sh -runjvmtools
```

Calling scripts with the jstack and jmap option can impact running jobs. Contact Teradata Services for recommendations before specifying.

Migrating Existing ARC Jobs to DSA

As of Data Mover 17.00, ARC is no longer a supported copy method. After upgrading to Data Mover 17.00 or later, jobs with ARC defined as the copy method no longer work. This includes jobs with the following options:

- Job forces the use of ARC as the force utility
- Job uses the `freeze job steps` option and the current job plan uses ARC

Teradata recommends using DSA as the replacement copy method for ARC. Support for DSA was added in Data Mover 16.20. Any existing ARC jobs can be migrated to DSA as part of the upgrade using the migration tool provided with the Data Mover 17.00 package or before the upgrade using the tool found at `/opt/teradata/datamover/support/arcmigration`. A readme file is provided with instructions on running the tool.

Migrating the Teradata Repository to the Postgres Repository

As of Data Mover 17.05, Data Mover switched the supported internal repository to a Postgres system. Existing Teradata repository data is automatically migrated to Postgres when upgrading to Data Mover 17.05 or later using the migration tool found at `/var/opt/teradata/postgres/migration/datamigration.sh`. A readme file is provided with instructions and requirements on running the tool if it needs to be run manually.

Configuring ActiveMQ on Remote Agents

In a high availability environment, ActiveMQ authentication is automatically configured when running `dmcluster config` during the install or upgrade process to Data Mover 17.00 or later. The designated-standby server and all additional agents in the clustered environment are automatically configured with the ActiveMQ secret and encrypted password from the designated-active Data Mover server.

However, if your environment is not configured to use a high availability cluster, perform the following steps after the install or upgrade so that any remote agents can successfully access ActiveMQ on the Primary server.

1. Copy `/etc/opt/teradata/tdactivemq/datamover.properties` from the Primary Data Mover server to the same location on the remote agent(s), as in the following example:

```
# scp /etc/opt/teradata/tdactivemq/datamover.properties
root@your_agent_server:/etc/opt/teradata/tdactivemq
```

2. Login to each remote agent to verify the ActiveMQ permissions are similar to the following:

```
-rw-r----- 1 root activemq 16 Feb 10 14:49 datamover.properties
```

3. If necessary, set the permissions for `datamover.properties` on each remote agent:

```
# chmod 640 datamover.properties
# chgrp activemq datamover.properties
```

4. Update the `agent.broker.password` property in the `agent.properties` file with the same value defined on the Primary Data Mover server.

Note:

Update `/etc/opt/teradata/datamover/agent.properties` if it exists. If not, update `opt/teradata/client/nn.nn/datamover/agent/agent.properties`.

5. Restart the agent.

```
/etc/init.d/dmagent stop
/etc/init.d/dmagent start
```



Upgrading Software

Upgrading Software

Upgrading Data Mover Software

Note:

All Data Mover upgrades except the DMCmdline package are performed by Teradata Customer Support.

To upgrade the Teradata® Data Mover software, do the following:

1. Create an incident on the Teradata Support Portal.
 2. Uninstall the DMCmdline package on your operating system.
 3. Reinstall the DMCmdline package on your operating system.
 4. Contact your Customer Service Representative.
-

Note:

- Make sure the DMFailover package from the monitoring servers is the same version as the Failover package from the Data Mover multi-purpose server.
 - When upgrading to Data Mover 17.10 or later from an older version, failover reconfiguration is mandatory. See [Configuring Automatic Failover](#).
-

Creating an Incident

You must obtain an Incident number from Teradata customer portal prior to performing any software upgrades.

1. Go to <https://support.teradata.com>.
2. Log in.
3. Click **Incidents** and follow the instructions.

Upgrading the Data Mover Command-Line Interface on Non-Teradata Servers

The Data Mover Command-Line Interface must be installed for Solaris Sparc, IBM AIX, Ubuntu, Windows systems, and Linux on non-Teradata servers using the following procedures. You cannot use PUT to install the Command-Line Interface on those systems.

Note:

If there is an existing installation on the system, it must be uninstalled before re-installing. You can have only one version of the Data Mover Command-Line package on a server.

Steps 1 through 4 do not apply to installation on Windows systems.

Only the major and minor versions of the Data Mover daemon and the Data Mover command line interface or Data Mover portlet must match.

1. Add the following lines of code to the end of the `/etc/profile` file to update the `JAVA_HOME` and `PATH` environment variables for all users:

```
export JAVA_HOME={full path of java installation location}
export PATH=$JAVA_HOME/bin:$PATH
```
2. Run the command:

```
source /etc/profile
```
3. Verify that the output shows JRE.1.8:

```
java -version
```
4. Open the `.profile` file of the root user and verify that the values for the `JAVA_HOME` and `PATH` environment variables are the same as those defined in `/etc/profile`.
 If the values are different, the `java -version` command will not produce the correct output during install time, and the installation will fail.
5. Copy the properties file to an outside directory if you want to preserve any customization that you made to the default values:

```
TDM_install_directory\CommandLine\commandline.properties
```
6. Uninstall and upgrade the appropriate software for your system as follows:

Operating System	Actions
Linux (for non-Teradata servers)	<ol style="list-style-type: none"> At the command line, type <code>export DM_INTERACTIVE_INSTALL=1</code> to set the environment variable for interactive install. At the command line, type the following: <pre>gunzip DMCmdline__linux_i386.17.12.xx.xx.tar.gz tar xvf DMCmdline__linux_i386.17.12.xx.xx.tar cd DMCmdline.17.12* rpm -Uvh DMCmdline__linux_noarch.17.12.xx.xx-1.rpm</pre> Answer the prompts as needed, and press Enter to accept the defaults where appropriate. Type <code>rpm -qa grep DMCmdline</code> to verify installation.
Solaris Sparc	<ol style="list-style-type: none"> At the command line, type <code>pkgrm DMCmdline</code> to uninstall. At the command line, type the following to upgrade: <pre>gunzip tdm-solaris__solaris_sparc.17.12.xx.xx.tar.gz tar xvf tdm-solaris__solaris_sparc.17.12.xx.xx.tar pkgadd -d 'pwd' DMCmdline</pre>

Operating System	Actions
	<ul style="list-style-type: none"> c. Answer the prompts as needed and press Enter to accept defaults where appropriate. d. Type <code>pkginfo -l DMCmdline</code> to verify installation.
IBM AIX	<ul style="list-style-type: none"> a. At the command line, type <code>installp -u DMCmdline</code> to uninstall. b. At the command line, type the following to upgrade: <code>gunzip tdm-aix__aix_power.17.12.xx.xx.tar.gz</code> <code>tar xvf tdm-aix__aix_power.17.12.xx.xx.tar</code> <code>installp -acF -d ./DMCmdline DMCmdline</code> c. Answer the prompts as needed and press Enter to accept defaults where appropriate. d. Type <code>ls1pp -l "DM*"</code> to verify installation.
Windows	<ul style="list-style-type: none"> a. To uninstall the existing DMCmdline software package, go to Start > Control Panel > Add or Remove Programs; then, select Teradata Data Mover Command Line Interface and select Remove. b. Copy the Data Mover directory on the media to a folder on the hard drive. c. Go to DataMover/Windows and unzip <code>tdm-windows__windows_i386.17.12.xx.xx.zip</code>. d. Go to the DISK1 directory and run <code>setup.exe</code>. e. Answer the prompts as needed and press Next to accept defaults where appropriate. f. Select Install when finished. g. Go to Start > Control Panel > Add or Remove Programs to verify installation.
Ubuntu	<ul style="list-style-type: none"> a. At the command line, type <code>dpkg -P dmcmdline</code> to uninstall. <ul style="list-style-type: none"> • The <code>commandline.properties</code> file is preserved as <code>commandline.properties.dpkgsave</code> in the <code>/opt/teradata/client/17.12/datamover/commandline</code> directory, and you can ignore the following warning: <pre>Warning: while removing dmcmdline, directory /opt/teradata /client/17.12/datamover/commandline is not empty so not removed</pre> • If you do not want to preserve the properties file, you can remove the <code>/opt/teradata/client/17.12/datamover/commandline</code> folder after uninstall is completed. b. At the command line, type the following: <code>tar xzvf tdm-ubuntu__ubuntu.17.12.xx.xx.tar.gz</code> <code>cd DMCmdline.17.12.xx.xx</code> <code>dpkg -i DMCmdline__ubuntu_all.17.12.xx.xx-1.deb</code> Note: In Ubuntu, <code>-i</code> is used for both install and upgrade. c. Type <code>dpkg -l grep dmcmdline</code> to verify the installation.

7. Restore the values from the properties file you copied to an outside directory if you want to preserve any customization that you made to the default values and override the values introduced by the patch

TDM_install_directory\CommandLine\commandline.properties

8. Specify the Data Mover REST server URL for communicating with the daemon as in the following example:

`dm.rest.endpoint=https://dm_host:1443/datamover`

Make sure the host:port value used for `dm.rest.endpoint` is on the `accept.host.list` in `tdmrest.properties`.

9. Log out of the current session and log back in to view the updated environment.

Upgrading the Data Mover Agent on a Linux Teradata Server

1. Copy the properties file to an outside directory if you want to preserve any customization that you made to the default values:

TDM_install_directory\agent\agent.properties

2. Uninstall and upgrade the appropriate software for your system as follows:

Operating System	Actions
Linux (for non-Teradata servers)	<ol style="list-style-type: none"> a. At the command line, type the following to upgrade the DMAgent and TTU packages: <code>./dminstallupgradeagent</code> b. Answer the prompts as needed, and press Enter to accept the defaults where appropriate. c. Type <code>rpm -qa grep DMAgent</code> to verify the installation.

3. Restore the values from the properties file you copied to an outside directory if you want to preserve any customization that you made to the default values and override the values introduced by the patch:
TDM_install_directory\Agent\agent.properties
4. Specify the broker URL and broker port number for communicating with the JMS bus.
The broker URL value is the machine name or IP address of the machine where ActiveMQ runs. The broker port value should also be the same as the port number that ActiveMQ uses. The defaults are `broker.url=localhost` and `broker.port=61616`.
5. Follow the steps in [Configuring ActiveMQ on Remote Agents](#) if this is the first time installing or upgrading a standalone agent using this script.
Once a standalone remote agent has been set up manually, the settings are preserved across upgrades.

Installing or Upgrading the Data Mover Portlet

You must install the Data Mover portlet package on the Teradata Viewpoint server to allow Teradata Viewpoint administrators configure and manage one or more Data Mover Daemons.

Note:

If there is an existing installation on the system, then uninstall it before re-installing. You can have only one version of the Data Mover portlet package on a Teradata Viewpoint server.

1. Use the compressed file `tdmportlets__linux_noarch.17.12.xx.xx-1.tar.gz` for Teradata Data Mover version 17.12.xx.xx portlets.
It contains the packages for the "Data Mover" and "Data Mover Setup" portlets that must be installed on the Teradata Viewpoint managed server.
2. For an existing version of the Teradata Data Mover portlets on the Viewpoint managed server, the portlet must be uninstalled before installing the new version.
3. You can only install the Teradata Data Mover portlets on the Viewpoint managed server as the root user.
4. Uncompress the `tdmportlets__linux_noarch.17.12.xx.xx-1.tar.gz` file using this command:

```
# gunzip -c tdmportlets__linux_noarch.17.12.xx.xx-1.tar.gz | tar xvf -
```
5. Uninstall the previous version of the Data Mover portlet using the following command:

```
# rpm -e tdmportlets
```
6. Install the Teradata Data Mover portlets by running the following command:

```
# rpm -Uvh tdmportlets__linux_noarch.17.12.xx.xx-1.rpm
```
7. Type the following at the command prompt: `# rpm -qa | grep tdmportlets`. If `tdmportlets-17.12.xx.xx-1` is included in the output then the Data Mover portlets were installed correctly.
8. Teradata recommends to restart all Viewpoint services after Data Mover portlet installation though viewpoint refreshes portlet libraries automatically without restart.

Additional Information

Audience

This guide is intended for use by:

- System administrators
- Database administrators and relational database developers
- Customers
- Teradata Customer Support

Changes and Additions

Date	Release	Description
September 2022	17.20.01.00	Added the capability of Cloud Staging Copy Service to use the password .txt file.
June 2022	17.20.00.00	<ul style="list-style-type: none"> • Introduced Cloud Staging Copy Service, a new feature for copying data through a cloud storage (AWS S3). • Added how to set up a cloud storage (AWS S3) called, Cloud Staging Area. • Added how to configure Cloud Staging Copy Service and DSC in Data Mover. • Described how to configure the Cloud Staging Copy REST Service. • Added how to simulate failover event.
January 2022	17.12.00.00	<ul style="list-style-type: none"> • Added a new demon setting for copying non-table objects. • Updated the commands for uncompressing and installing the Teradata Data Mover portlet files. • Added the system requirements for the Data Mover server across all cloud platforms.
September 2021	17.11.00.00	<ul style="list-style-type: none"> • Added information on upgrading the Data Mover Portlet on a Viewpoint server • Added Single sign-on support
June 2021	17.10.00.00	<ul style="list-style-type: none"> • Added information on TLS 1.2 Data Path Encryption for DSC on the Data Mover server • Added information on TLS 1.2 Data Path Encryption for Standby DSC • Updated DSA Failover Configurations • Added information on TLSv1.2 configuration for DSA Jobs

Teradata Links

Link	Description
https://docs.teradata.com/	Search Teradata Documentation, customize content to your needs, and download PDFs. Customers: Log in to access Orange Books.
https://support.teradata.com	Helpful resources in one place: <ul style="list-style-type: none"> • Support requests • Account management and software downloads • Knowledge base, community, and support policies • Product documentation • Learning resources, including Teradata University
https://www.teradata.com/University/Overview	Teradata education network
https://support.teradata.com/community	Link to Teradata community

Related Documentation

Title	Publication ID
<i>Teradata® Data Mover User Guide</i> Describes how to use the Teradata Data Mover portlets and command-line interface.	B035-4101
<i>Parallel Upgrade Tool (PUT) Reference</i> Describes how to install application software using PUT.	B035-5713
<i>Teradata® Viewpoint User Guide</i> Describes the Teradata Viewpoint portal, portlets, and system administration features.	B035-2206

Customer Education

Teradata Customer Education delivers training for your global workforce, including scheduled public courses, customized on-site training, and web-based training. For information about the classes, schedules, and the Teradata Certification Program, go to <https://www.teradata.com/TEN/>.

Customer Support

Customer support is available around-the-clock, seven days a week through the Global Technical Support Center (GSC). To learn more, go to <https://support.teradata.com>.