Informatica®

Informatica® Cloud Data Integration

Microsoft SQL Server Connector

Informatica Cloud Data Integration Microsoft SQL Server Connector
July 2024

© Copyright Informatica LLC 2017, 2024

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, Informatica Cloud, and PowerCenter are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at https://www.informatica.com/trademarks.html. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

See patents at https://www.informatica.com/legal/patents.html.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2024-07-15

Table of Contents

Preface 5

Informatica Resources 5

- Informatica Documentation 5
- Informatica Intelligent Cloud Services web site 5
- Informatica Intelligent Cloud Services Communities 5
- Informatica Intelligent Cloud Services Marketplace 6
- Data Integration connector documentation 6
- Informatica Knowledge Base 6
- Informatica Intelligent Cloud Services Trust Center 6
- Informatica Global Customer Support 6

Chapter 1: Introduction to Microsoft SQL Server Connector 7

Microsoft SQL Server Connector assets 7

Chapter 2: Connections for Microsoft SQL Server 8

Prepare for authentication 8

- Prepare for Kerberos authentication 8

Connect to Microsoft SQL Server 10

- Before you begin 10
- Connection details 10
- Authentication types 10
- Advanced settings 15

Configure SSL with the serverless runtime environment 16

Chapter 3: Synchronization tasks with Microsoft SQL Server Connector 18

Microsoft SQL Server sources in synchronization tasks 18

Microsoft SQL Server targets in synchronization tasks 19

Advanced properties for Microsoft SQL Server sources and targets in synchronization tasks 19

Microsoft SQL Server lookup in synchronization tasks 20

Synchronization task example 20

Chapter 4: Mappings and mapping tasks with Microsoft SQL Server Connector 23

Microsoft SQL Server sources in a mapping 23

Key range partitioning 25

- Configuring key range partitioning 26

Pass-through partitioning 26

- Processing Unicode characters 26

Microsoft SQL Server targets in a mapping 27

- Configuring an update override for the target 28

Microsoft SQL Server lookups in mapping. 29
Microsoft SQL Server mapping task example. 30
Rules and guidelines for Microsoft SQL Server tasks. 32
SQL ELT optimization. 33
Full SQL ELT optimization. 34
Source SQL ELT optimization. 34
SQL ELT optimization functions. 34
SQL ELT optimization variables. 37
Configuring SQL ELT optimization. 37
Cross-schema SQL ELT optimization. 37
Rules and guidelines for SQL ELT optimization. 38

Chapter 5: Replication tasks with Microsoft SQL Server. 40
Microsoft SQL Server sources in replication tasks. 41
Microsoft SQL Server targets in replication tasks. 41

Chapter 6: Data type reference. 42
Microsoft SQL Server and transformation data types. 42

Index. 45

Preface

Use *Microsoft SQL Server Connector* to learn how to read from or write to Microsoft SQL Server by using Cloud Data Integration. Learn to create a connection, develop mappings, and run synchronization, mapping, dynamic mapping, replication, and data transfer tasks. You can also learn how to configure SQL ELT optimization and cross-schema SQL ELT optimization to a Microsoft SQL Server database.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Informatica Cloud integration services.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

You can contact a Global Support Center through the Informatica Network or by telephone.

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to Microsoft SQL Server Connector

You can use Microsoft SQL Server Connector to connect to Microsoft SQL Server databases from Data Integration. You can read data from or write data to the Microsoft SQL Server databases.

You can also use a Microsoft SQL Server connection to connect to the following databases:

- Microsoft Azure SQL Database
- Audit-enabled Azure SQL Database and pre-V12 Azure SQL Database
- Azure SQL Database from Secure Agents deployed on Azure virtual machines
- Azure SQL Database managed instance
- Amazon RDS SQL Server and Azure SQL Server private endpoints on virtual networks

You can switch mappings to advanced mode to include transformations and functions that enable advanced functionality.

Microsoft SQL Server Connector assets

Create assets in Data Integration to integrate data using Microsoft SQL Server Connector.

When you use Microsoft SQL Server Connector, you can include the following Data Integration assets:

- Data transfer task
- Dynamic mapping task
- Mapping
- Mapping task
- PowerCenter task
- Replication task
- Synchronization task

For more information about configuring assets and transformations, see *Mappings*, *Transformations*, and *Tasks* in the Data Integration documentation.

CHAPTER 2

Connections for Microsoft SQL Server

Create a connection to read from or write to Microsoft SQL Server or Microsoft Azure SQL Database.

You can use a Microsoft SQL Server connection to specify sources, targets, and lookups in mappings and tasks.

Prepare for authentication

You can configure Database or Kerberos authentication method to connect to Microsoft SQL Server. Before you configure the connection properties, you need to keep the authentication details handy based on the authentication type that you want to use. For Kerberos authentication, you need to configure certain prerequisites.

Prepare for Kerberos authentication

You can use Kerberos authentication to connect to Microsoft SQL Server databases by placing the required configuration files on the Secure Agent machine. You can also use Kerberos authentication to connect to SSL-enabled Microsoft SQL Server databases.

When you configure Kerberos authentication to connect to Microsoft SQL Server, consider the following guidelines:

- You can't use the Hosted Agent or serverless runtime environment.
- Ensure that the Secure Agent and database server that you use are registered in the KDC server.
- You can't add more than one KDC to a krb5.conf file.
- You can't generate a credential cache file for more than one Kerberos principal user.

Configuring Kerberos authentication

Before you use Kerberos authentication to connect to Microsoft SQL Server on Linux or Windows, the organization administrator needs to perform the prerequisite tasks.

1. To configure the Java Authentication and Authorization Service configuration file (JAAS), perform the following tasks:
   a. Create a JAAS configuration file on the Secure Agent machine.

b. Add the following entries to the JAAS configuration file:

```
JDBC_DRIVER_01 {
com.sun.security.auth.module.Krb5LoginModule required useTicketCache=true;
};
```

2. To configure the `krb5.conf` file, perform the following tasks:
   a. Create a `krb5.conf` file on the Secure Agent machine.
   b. Add the details of the Key Distribution Center (KDC) and admin server to the `krb5.conf` file in the following format:

```
[libdefaults]
default_realm = <Realm name>
forwardable = true
ticket_lifetime = 24h

[realms]
<REALM NAME> = {
kdc = <Location where KDC is installed>
admin_server = <Location where KDC is installed>
                                    }
 [domain_realm]
<domain name or host name> = <Domain name or host name of Kerberos>
<domain name or host name> = <Domain name or host name of Kerberos>
```

3. Set the following environment variables on the Secure Agent machine.
   For more information about the required environment variables, see "Setting environment variables" on page 9.
4. Restart the Secure Agent.
5. To generate the credential cache file on the Secure Agent machine and use Kerberos authentication to connect to Microsoft SQL Server, perform the following tasks:
   a. On the Secure Agent machine, run the following command and specify the Microsoft SQL Server user name and realm name:

```
Kinit <user name>@<realm_name>
```

   b. When prompted, enter the password for the Kerberos principal user.

Setting environment variables

To use Kerberos authentication to connect to Microsoft SQL Server, you need to set the required environment variables on the Secure Agent machine.

Set the following environment variables:

- `setenv KRB5CCNAME <Absolute path and file name of the credentials cache file>`
- `setenv KRB5_CONFIG <Absolute path of the Kerberos configuration file>\krb5.conf`
- `setenv JAASCONFIG <Absolute path of the JAAS config file>\<File name>.conf`

After you set the environmental variables, you need to restart the Secure Agent.

Alternatively, you can add the environment variables when you create a Microsoft SQL Server connection.

To add the environment variables when you configure a connection and use Kerberos authentication, you need to add the *KRB5_CONFIG*, *KRB5CCNAME*, and *JAASCONFIG* properties in the **Metadata Advanced Connection Properties** field in a Microsoft SQL Server connection.

For example, add the properties in the following format:

```
KRB5_CONFIG=<Absolute path of the Kerberos configuration file>
\krb5.conf;KRB5CCNAME=<Absolute path of the credential cache file>/<File
name>;JAASCONFIG=<Absolute path of the JAAS config file>\<File name>.conf
```

Note: Ensure that you separate each key-value pair with a semicolon.

Connect to Microsoft SQL Server

Let's configure the Microsoft SQL Server connection properties to connect to Microsoft SQL Server databases.

Before you begin

Before you get started, you'll need to get information from your SQL Server DB account based on the authentication method and the type of SQL server DB to which you want to connect.

Check out "Prepare for authentication" on page 8 to learn more about the authentication prerequisites.

Connection details

The following table describes the Microsoft SQL Server connection properties:

Property	Description
Connection Name	Name of the connection. Each connection name must be unique within the organization. Connection names can contain alphanumeric characters, spaces, and the following special characters: _ . + -, Maximum length is 255 characters.
Description	Description of the connection. Maximum length is 4000 characters.
Type	Type of connection. Select SQL Server from the list.
Runtime Environment	The name of the runtime environment where you want to run the tasks. Specify a Secure Agent, Hosted Agent, or serverless runtime environment.
SQL Server Version	Microsoft SQL Server database version.

Authentication types

You can configure one of the following authentication modes to connect to Microsoft SQL Server databases:

- SQL Server Authentication. Uses your Microsoft SQL Server user name and password to access Microsoft SQL Server.
- Windows Authentication (Deprecated). Uses the Microsoft Windows authentication to access Microsoft SQL Server. This option is available when you access Data Integration by using Microsoft Windows.

  When you choose this option, you don't need to enter credentials to access Microsoft SQL Server and ensure that the user account that starts the Secure Agent service is available in the Microsoft SQL Server database.

  Note: Windows authentication is not certified for Microsoft SQL Server 2017 version hosted on Linux. You can't configure Windows Authentication when you use a serverless runtime environment.

- Active Directory Password. Uses the Azure Active Directory user name and password to authenticate and access the Microsoft Azure SQL Database.
- Windows Authentication v2. Uses this authentication method to access Microsoft SQL Server from Data Integration using an agent hosted on a Linux or Windows machine.
  When you choose this option on Linux, enter your domain name and Microsoft Windows credentials to access Microsoft SQL Server.

  When you choose this option on Windows, the agent uses the user credentials specified in the connection only to test the connection. During runtime, the agent uses the credentials of the user who started the Secure Agent service. Ensure that the user account that starts the Secure Agent service is available in the Microsoft SQL Server database.

  Note: You can't configure Windows Authentication when you use a serverless runtime environment.
- Kerberos. Uses Kerberos authentication to connect to Microsoft SQL Server.
  When you choose this option on Windows, ensure that the user account that starts the Secure Agent service is available in the Microsoft SQL Server database. You don't need to enter your credentials to access Microsoft SQL Server.

  Note: You can't configure the Kerberos authentication when you use a Hosted Agent or serverless runtime environment.

Select the required authentication type and then configure the authentication-specific parameters.

Default is SQL Server Authentication.

SQL Server authentication

The following table describes the basic connection properties for SQL Server authentication:

Property	Description
Domain	Applies to Windows Authentication v2. The domain name of the Windows user.
User Name	User name for the database login. The user name can't contain a semicolon. To connect to Microsoft Azure SQL Database, specify the user name in the following format: `username@host` If you use Windows Authentication v2 on Windows, the user name is used as follows: - During design time, the agent uses the user name specified here to test the connection. - During runtime, the Microsoft SQL server driver ignores the user name specified in this field and uses the credentials of the user who started the Secure Agent service. If you use Windows Authentication v2 on Linux, the user name specified here is used both during design time and runtime. **Note:** This property is not applicable if you use the Windows Authentication mode to access Microsoft SQL Server.
Password	Password for the database login. The password can't contain a semicolon. If you use Windows Authentication v2 on Windows, the password is used as follows: - During design time, the agent uses the password specified here to test the connection. - During runtime, the Microsoft SQL server driver ignores the password specified in this field and uses the credentials of the user who started the Secure Agent service. If you use Windows Authentication v2 on Linux, the password specified here is used both during design time and runtime. **Note:** This property is not applicable if you use the Windows Authentication mode to access Microsoft SQL Server.

Property	Description
Host	Name of the machine hosting the database server. To connect to Microsoft Azure SQL Database, specify the fully qualified host name. For example, `vmjcmwxsfboheng.westus.cloudapp.azure.com`.
Port	Network port number used to connect to the database server. Default is 1433.
Instance Name	Instance name of the Microsoft SQL Server database.
Database Name	Database name for the Microsoft SQL Server target connection. Database name is case-sensitive if the database is case-sensitive. Maximum length is 100 characters. Database names can include alphanumeric and underscore characters.
Schema	Schema used for the target connection.
Code Page	The code page of the database server.

Windows authentication

The following table describes the basic connection properties for Windows authentication:

Property	Description
Domain	Applies to Windows Authentication v2. The domain name of the Windows user.
Host	Name of the machine hosting the database server. To connect to Microsoft Azure SQL Database, specify the fully qualified host name. For example, `vmjcmwxsfboheng.westus.cloudapp.azure.com`.
Port	Network port number used to connect to the database server. Default is 1433.
Instance Name	Instance name of the Microsoft SQL Server database.
Database Name	Database name for the Microsoft SQL Server target connection. Database name is case-sensitive if the database is case-sensitive. Maximum length is 100 characters. Database names can include alphanumeric and underscore characters.
Schema	Schema used for the target connection.
Code Page	The code page of the database server.

Active Directory Password authentication

The following table describes the basic connection properties for Active Directory Password authentication:

Property	Description
Domain	Applies to Windows Authentication v2. The domain name of the Windows user.
User Name	User name for the database login. The user name can't contain a semicolon. To connect to Microsoft Azure SQL Database, specify the user name in the following format: `username@host` If you use Windows Authentication v2 on Windows, the user name is used as follows: - During design time, the agent uses the user name specified here to test the connection. - During runtime, the Microsoft SQL server driver ignores the user name specified in this field and uses the credentials of the user who started the Secure Agent service. If you use Windows Authentication v2 on Linux, the user name specified here is used both during design time and runtime. **Note:** This property is not applicable if you use the Windows Authentication mode to access Microsoft SQL Server.
Password	Password for the database login. The password can't contain a semicolon. If you use Windows Authentication v2 on Windows, the password is used as follows: - During design time, the agent uses the password specified here to test the connection. - During runtime, the Microsoft SQL server driver ignores the password specified in this field and uses the credentials of the user who started the Secure Agent service. If you use Windows Authentication v2 on Linux, the password specified here is used both during design time and runtime. **Note:** This property is not applicable if you use the Windows Authentication mode to access Microsoft SQL Server.
Host	Name of the machine hosting the database server. To connect to Microsoft Azure SQL Database, specify the fully qualified host name. For example, `vmjcmwxsfboheng.westus.cloudapp.azure.com`.
Port	Network port number used to connect to the database server. Default is 1433.
Instance Name	Instance name of the Microsoft SQL Server database.
Database Name	Database name for the Microsoft SQL Server target connection. Database name is case-sensitive if the database is case-sensitive. Maximum length is 100 characters. Database names can include alphanumeric and underscore characters.
Schema	Schema used for the target connection.
Code Page	The code page of the database server.

Windows Authentication V2

The following table describes the basic connection properties for Windows Authentication V2:

Property	Description
Domain	Applies to Windows Authentication v2. The domain name of the Windows user.
User Name	User name for the database login. The user name can't contain a semicolon. To connect to Microsoft Azure SQL Database, specify the user name in the following format: `username@host` If you use Windows Authentication v2 on Windows, the user name is used as follows: - During design time, the agent uses the user name specified here to test the connection. - During runtime, the Microsoft SQL server driver ignores the user name specified in this field and uses the credentials of the user who started the Secure Agent service. If you use Windows Authentication v2 on Linux, the user name specified here is used both during design time and runtime. **Note:** This property is not applicable if you use the Windows Authentication mode to access Microsoft SQL Server.
Password	Password for the database login. The password can't contain a semicolon. If you use Windows Authentication v2 on Windows, the password is used as follows: - During design time, the agent uses the password specified here to test the connection. - During runtime, the Microsoft SQL server driver ignores the password specified in this field and uses the credentials of the user who started the Secure Agent service. If you use Windows Authentication v2 on Linux, the password specified here is used both during design time and runtime. **Note:** This property is not applicable if you use the Windows Authentication mode to access Microsoft SQL Server.
Host	Name of the machine hosting the database server. To connect to Microsoft Azure SQL Database, specify the fully qualified host name. For example, `vmjcmwxsfboheng.westus.cloudapp.azure.com`.
Port	Network port number used to connect to the database server. Default is 1433.
Instance Name	Instance name of the Microsoft SQL Server database.
Database Name	Database name for the Microsoft SQL Server target connection. Database name is case-sensitive if the database is case-sensitive. Maximum length is 100 characters. Database names can include alphanumeric and underscore characters.
Schema	Schema used for the target connection.
Code Page	The code page of the database server.

Kerberos authentication

The following table describes the basic connection properties for Kerberos authentication:

Property	Description
Domain	Applies to Windows Authentication v2. The domain name of the Windows user.
Host	Name of the machine hosting the database server. To connect to Microsoft Azure SQL Database, specify the fully qualified host name. For example, `vmjcmwxsfboheng.westus.cloudapp.azure.com`.
Port	Network port number used to connect to the database server. Default is 1433.
Instance Name	Instance name of the Microsoft SQL Server database.
Database Name	Database name for the Microsoft SQL Server target connection. Database name is case-sensitive if the database is case-sensitive. Maximum length is 100 characters. Database names can include alphanumeric and underscore characters.
Schema	Schema used for the target connection.
Code Page	The code page of the database server.

Advanced settings

The following table describes the advanced connection properties:

Property	Description
Encryption Method	The method that the Secure Agent uses to encrypt the data sent between the driver and the database server. You can use the encryption method to connect to Microsoft Azure SQL Database. Default is None.
Crypto Protocol Version	Cryptographic protocols to use when you enable SSL encryption.
Validate Server Certificate	When set to True, Secure Agent validates the certificate that is sent by the database server. If you specify the HostNameInCertificate parameter, Secure Agent also validates the host name in the certificate. When set to false, the Secure Agent doesn't validate the certificate that is sent by the database server.
Trust Store	The location and name of the truststore file. The truststore file contains a list of Certificate Authorities (CAs) that the driver uses for SSL server authentication. For the serverless runtime environment, specify the following certificate path in the serverless agent directory: `/home/cldagnt/SystemAgent/serverless/configurations/ssl_store/<TrustStore_filename>`

Property	Description
Trust Store Password	The password to access the contents of the truststore file.
Host Name in Certificate	Host name of the machine that hosts the secure database. If you specify a host name, the Secure Agent validates the host name included in the connection with the host name in the SSL certificate.

Metadata Advanced Connection Properties	Additional properties for the JDBC driver to fetch the metadata.
	Enter properties in the following format:
	`<parameter name>=<parameter value>`
	If you enter more than one property, separate each key-value pair with a semicolon.
	For example, enter the following property to configure the connection timeout when you test a connection:
	`LoginTimeout=<value_in_seconds>`
	Note: The default connection timeout is 270 seconds.

Runtime Advanced Connection Properties	Additional properties for the ODBC driver required at run time. If you specify more than one property, separate each key-value pair with a semicolon.

Configure SSL with the serverless runtime environment

You can use the serverless runtime environment with Microsoft SQL Server Connector to connect to an SSL-enabled Microsoft SQL Server database.

Before you configure a secure Microsoft SQL Server connection using the serverless runtime environment, complete the following prerequisite tasks to add the SSL certificates to the serverless runtime location:

1. Create the following structure for the serverless agent configuration in AWS or Azure: `<Supplementary file location>/serverless_agent_config`
2. Add the truststore certificate in the Amazon S3 bucket or Azure container in the following location in your AWS or Azure account: `<Supplementary file location>/serverless_agent_config/SSL`
3. Copy the following code snippet to a text editor:

```
version: 1
agent:
  agentAutoApply:
    general:
      sslStore:
        - fileCopy:
            sourcePath: SSL/<TrustStore_filename>
```

   where the source path is the directory path of the certificate file in AWS or Azure.
4. Ensure that the syntax and indentations are valid, and then save the file as `serverlessUserAgentConfig.yml` in the following AWS or Azure location: `<Supplementary file location>/serverless_agent_config`

When the .yml file runs, the SSL certificate is copied from the AWS or Azure location to the serverless agent directory.

5. In the Microsoft SQL Server connection properties, specify the following certificate path in the serverless agent directory in the **Trust Store** field: `/home/cldagnt/SystemAgent/serverless/configurations/ssl_store/<cert_filename>`

CHAPTER 3

Synchronization tasks with Microsoft SQL Server Connector

Use a Synchronization task to synchronize data between a source and target.

You can configure a synchronization task using the Synchronization Task wizard. You can use Microsoft SQL Server objects as sources, targets, or lookup objects.

When you create a task, you can associate it with a schedule to run it at specified times or on regular intervals. Or, you can run it manually. You can monitor tasks that are currently running in the activity monitor and view logs about completed tasks in the activity log.

Microsoft SQL Server sources in synchronization tasks

When you configure a synchronization task to use a Microsoft SQL Server source, you can read data from a single object or multiple related objects. To read data from multiple sources, you must define relationships based on key columns or create a user-defined join condition.

To optimize performance, you can configure a filter in the **Data Filters** tab where you can select an object, field, operator, and enter a filter value based on which the synchronization task runs. You cannot include UTF-8 characters in a data filter on a Microsoft SQL Server source.

You can also configure an advanced filter to define a more complex filter condition, which can include multiple conditions using the AND or OR logical operators.

The following table describes the Microsoft SQL Server source properties:

Property	Description
Connection	Name of the active source connection.
Source Type	Type of the source object available. You can choose from the following source types: - Single - Multiple - Saved query

Property	Description
Source Object	Name of the source object.
Display source fields in alphabetical order	Displays source fields in alphabetical order. By default, fields appear in the order returned by the source system.

Microsoft SQL Server targets in synchronization tasks

You can use a single object as a target in a synchronization task.

The following table describes the Microsoft SQL Server target properties:

Property	Description
Connection	Name of the active target connection.
Target Object	Name of target objects available in the connection. Select an existing object from the list or create a new object at the run time.
Truncate Target	When you enable the **Truncate Target** option, the Secure Agent truncates table before writing data to target. Default is false.
Enable Target Bulk Load	Uses Microsoft SQL Server bulk API to write data in bulk mode to Microsoft SQL Server.
Display target fields in alphabetical order	Displays source fields in alphabetical order. By default, fields appear in the order returned by the source system.

Advanced properties for Microsoft SQL Server sources and targets in synchronization tasks

When you configure a synchronization task to use a Microsoft SQL Server source or target, you can configure advanced properties. Advanced properties appear on the **Schedule** page of the Synchronization Task wizard.

The following table describes the advanced properties that you can configure for a Microsoft SQL Server source or target:

Property	Description
Preprocessing Commands	Enter the pre-processing script that should be executed before running the synchronization task.
Post-processing Commands	Enter the post-processing script that should be executed after running the synchronization task.

Property	Description
Parameter File Name	Name of the file that contains the parameters to be used in filters or expressions.
Maximum Number of Log Files	Enter the maximum number of log files that the Secure Agent generates until the number reaches the limit that you set. Default is 10.

Microsoft SQL Server lookup in synchronization tasks

When you configure field mappings in a synchronization task, you can create a lookup to a Microsoft SQL Server object. When you use a Microsoft SQL Server object as a lookup, you do not need to configure specific Microsoft SQL Server properties.

Synchronization task example

To perform the insert operation where you can read data from a Microsoft SQL Server Source and can write data to the Microsoft SQL Server target.

Perform the following steps to create a synchronization task::

Note: You need to create a Microsoft SQL Server connection before configuring the synchronization task.

1. In Data Integration, click **New** > **Tasks**.
2. Select **Synchronization Task**, and click **Create**.

   The **Definition** tab appears.
3. Configure the following fields on the **Definition** tab:

Field	Description
Task Name	Name of the synchronization task.
Description	Description of the synchronization task. Maximum length is 255 characters.
Task Operation	Select the task operation that you want to perform.

4. Click **Next**.

   The **Source** tab appears.

5. Configure the following fields on the **Source** tab:

Field	Description
Connection	Select a Microsoft SQL Server connection you created.
Source Type	Select the source type. By default, **Single** is selected.
Source Object	Select the required object from the list.
Display technical names instead of labels	Displays technical names instead of business names.
Display source fields in alphabetical order	Displays source fields in alphabetical order. By default, fields appear in the order returned by the source system.

6. Click **Next**.

   The **Target** tab appears.

7. Configure the following fields on the **Target** tab:

Field	Description
Connection	Name of the active target connection.
Target Object	Name of target objects available in the connection. Select an existing object from the list or create a new object at the run time.
Truncate Target	When you enable the **Truncate Target** option, the Secure Agent truncates table before writing data to target. Default is false.
Enable Target Bulk Load	Uses Microsoft SQL Server bulk API to write data in bulk mode to Microsoft SQL Server.
Display target fields in alphabetical order	Displays target fields in alphabetical order. By default, fields appear in the order returned by the target system.

8. Click **Next**.

   The **Data Filters** tab appears.

9. Select the filter object, filter field, and filter operator to create a data filter on the **Data Filters** page.
10. Click **Next**.

    The **Field Mapping** tab appears.

11. Click **Automatch** on the **Field Mapping** tab to map source fields to target fields accordingly.
12. Click **Validate Mapping** to validate the mapping.
13. Click **Next**.

    The **Schedule** tab appears where you can schedule the task for each requirement and save.

14. Configure the advanced source properties on the **Schedule** tab.
15. Click **Save** > **Finish**.

16. Select the synchronization task from the **Explore** page and click **Actions** > **Run**.
    In Monitor, you can monitor the status of the logs after you run the task.

CHAPTER 4

Mappings and mapping tasks with Microsoft SQL Server Connector

Use the Data Integration Mapping Designer to create a mapping. When you create a mapping, you configure a source or target to represent a Microsoft SQL Server object.

In advanced mode, the Mapping Designer updates the mapping canvas to include transformations and functions that enable advanced functionality.

Describe the flow of data from source and target along with the required transformations before the agent writes data to the target. When you create a mapping task, select the mapping that you want to use. Use the Mapping Task wizard to create a mapping task. Validate and run the mapping to read data from sources and write to a target. The mapping task processes data based on the data flow logic you define in the mapping.

Microsoft SQL Server sources in a mapping

To read data from a Microsoft SQL Server database, configure a Microsoft SQL Server object as the Source transformation in a mapping. Enable partitioning when you configure the Source transformation in the Mapping Designer to optimize the performance of the mapping task. You can use the source or full SQL ELT optimization to read data from Microsoft SQL server.

Specify the name and description of the Microsoft SQL Server source. Configure the source, query options, and advanced properties for the source object.

The following table describes the source properties that you can configure for a Microsoft SQL Server source:

Property	Description
Connection	Name of the source connection. You can select an existing connection, create a new connection, or define parameter values for the source connection property. If you want to overwrite the source connection properties at runtime, select the **Allow parameter to be overridden at run time** option. Specify the parameter file directory and name in the advanced session properties.
Source Type	Type of the Microsoft SQL Server source object available. You can choose from the following source types: - Single - Multiple - Query - Parameter
Parameter	A parameter file where you define values that you want to update without having to edit the task. Select an existing parameter for the source object or click **New Parameter** to define a new parameter for the source object. The **Parameter** property appears only if you select parameter as the source type. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option. When the task runs, the Secure Agent uses the parameters from the file that you specify in the advanced session properties.
Object	Name of the Microsoft SQL Server source object.
Filter	Configure a simple filter or an advanced filter to remove rows at the source. You can improve efficiency by filtering early in the data flow. A simple filter includes a field name, operator, and value. Use an advanced filter to define a more complex filter condition, which can include multiple conditions using the AND or OR logical operators. When you configure partitioning, separate multiple filter conditions with parentheses ().
Sort	Select the fields and type of sorting to use. To sort data for a parameterized source, you must use a parameter for the sort options.
Select distinct rows	Select this option to extract only distinct rows.

The following table describes the advanced source properties that you can configure for a Microsoft SQL Server source:

Property	Description
Tracing level	Amount of detail that appears in the log for this transformation. You can choose terse, normal, verbose initialization, or verbose data. Default is normal.
Pre SQL	Pre-SQL command that must be run before reading data from the source.
Post SQL	Post-SQL command that must be run after reading data from the source.

Property	Description
SQL Override	The SQL statement to override the default query generated from the specified source type to read data from the Microsoft SQL Server source. You must ensure that the number of columns, order of columns, and the data type you specify in the query matches with the Microsoft SQL Server source object.
Output is Deterministic	Relational source or transformation output that does not change between session runs when the input data is consistent between runs. When you configure this property, the Secure Agent does not stage source data for recovery if transformations in the pipeline always produce repeatable data.
Output is repeatable	Relational source or transformation output that is in the same order between session runs when the order of the input data is consistent. When output is deterministic and output is repeatable, the Secure Agent does not stage source data for recovery.

Key range partitioning

You can configure key range partitioning when you use a mapping task to read data from Microsoft SQL Server sources. With key range partitioning, the Secure Agent distributes rows of source data based on the field that you define as partition keys. The Secure Agent compares the field value to the range values for each partition and sends rows to the appropriate partitions.

Use key range partitioning for columns that have an even distribution of data values. Otherwise, the partitions might have unequal size. For example, a column might have 10 rows between key values 1 and 1000 and the column might have 999 rows between key values 1001 and 2000. If the mapping includes multiple sources, use the same number of key ranges for each source.

When you define key range partitioning for a column, the Secure Agent reads the rows that are within the specified partition range. For example, if you configure two partitions for a column with the ranges as 10 through 20 and 30 through 40, the Secure Agent does not read the rows 20 through 30 because these rows are not within the specified partition range.

Consider the following rules when you configure partitioning:

- You can configure a partition key for fields of the following data types:
  - String
  - Any type of number data type. However, you cannot use decimals in key range values.
  - Date/time type. Use the following format: MM/DD/YYYY HH24:MI:SS
- You cannot use key range partitions when a mapping includes any of the following transformations:
  - Web Services
  - XML to Relational
- When you specify the datetime and datetime2 columns with subseconds as the key range values in a partitioning, the records are not loaded to the target. You must delete the value in subseconds from the values specified in the partitioning and run the mapping.

Configuring key range partitioning

Perform the following steps to configure key range partitioning for Microsoft SQL Server sources:

1. In the Source Properties, click the **Partitions** tab.
2. Select the required partition key from the list.
3. Click **Add New Key Range** to define the number of partitions and the key ranges based on which the Secure Agent must partition data.

   Use a blank value for the start range to indicate the minimum value. Use a blank value for the end range to indicate the maximum value.

Pass-through partitioning

You can configure pass-through partitioning to optimize the performance of mapping tasks when you read data from Microsoft SQL Server sources.

To configure pass-through partitioning, specify a query or filter value for each partition to read data in parallel.

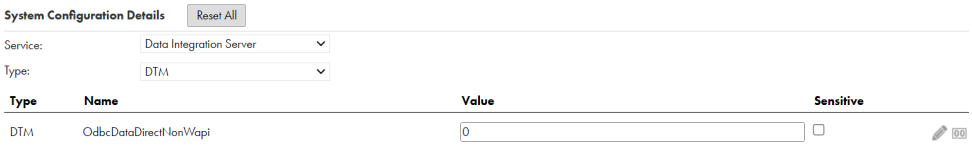
Consider the following rules and guidelines when you run a mapping task enabled for pass-through partitioning to read from Microsoft SQL Server:

- When you upsert data in a Microsoft SQL Server target, data discrepancies occur.
- Even though you configure a filter in the Source transformation, you can't filter source data.
- When you specify an SQL override in a Source transformation, the SQL override statement overrides the SQL query and source filter statements for the configured partitions.
  For example, if you specify an SQL query or source filter for five partitions, the SQL override overrides the SQL query and source filter statements for all the five partitions.

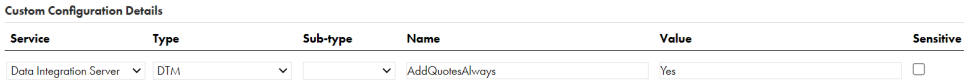
Processing Unicode characters

Before you enable pass-through partitioning in a mapping task when the SQL Query or Source Filter value of the configured partitions contain Unicode characters, you need to set certain system and custom properties for the Secure Agent in Administrator.

1. Select **Administrator** > **Runtime Environments**.
2. On the **Runtime Environments** page, select the Secure Agent that runs the mapping task.
3. Click **Edit**.
4. In the **System Configuration Details** section, perform the following steps:
   a. Select **Data Integration Server** as the service and **DTM** as the type.
   b. Edit the `OdbcDataDirectNonWapi` system property and set the value to **0**.



5. In the **Custom Configuration Details** section, perform the following steps:
   a. Select **Data Integration Server** as the service and **DTM** as the type.
   b. Enter `AddQuotesAlways` in the **Name** field and set the value to **Yes**.



6. Click **Save**.

Microsoft SQL Server targets in a mapping

To write data to a Microsoft SQL Server database, configure a Microsoft SQL Server object as the Target transformation in a mapping. You can use the full SQL ELT optimization to write data to Microsoft SQL Server.

Specify the name and description of the Microsoft SQL Server target. Configure the target and advanced properties for the target object.

The following table describes the target properties that you can configure for a Microsoft SQL Server target:

Property	Description
Connection	Name of the target connection. You can select an existing connection, create a new connection, or define parameter values for the target connection property. If you want to overwrite the target connection properties at runtime, select the **Allow parameter to be overridden at run time** option.
Target Type	Type of the Microsoft SQL Server target object available. You can choose from the following source types: - Single Object - Parameter
Parameter	A parameter file where you define values that you want to update without having to edit the task. Select an existing parameter for the target object or click **New Parameter** to define a new parameter for the target object. The **Parameter** property appears only if you select parameter as the target type. If you want to overwrite the target object at runtime, select the **Allow parameter to be overridden at run time** option. When the task runs, the Secure Agent uses the parameters from the file that you specify in the advanced session properties.
Object	Name of the Microsoft SQL Server target object based on target type selected. Specify the target object that you want to create at run time. You can also select an existing object from the list.

Property	Description
Operation	Select the target operation. You can perform the following operations on a Microsoft SQL Server target: - Insert - Update - Upsert (Update or Insert) - Delete - Data Driven
Data Driven Condition	Enables you to define expressions that flag rows for an insert, update, or delete operation. Appears only when the operation type is **Data Driven**. **Note:** To configure a data driven expression in a mapping, you need to specify the expression that uses the IIF function.
Update Columns	Click **Add** to select the columns that you want to use as a logical primary key for performing update, upsert, delete, and data driven operations on the target. This field is not required if the target table already has a primary key.
Truncate target	When you enable the **Truncate Target** option, the Secure Agent truncates the table before running the task. Default is not selected.
Enable target bulk load	Uses Microsoft SQL Server bulk API to insert data in bulk mode. When you create a mapping task, you can use the Microsoft SQL Server bulk API to perform insert operation.
Forward Rejected Rows	Determines whether the transformation passes rejected rows to the next transformation or drops rejected rows. If you select the option, the Secure Agent flags the rows for reject and writes them to the reject file. If you do not select the option, the Secure Agent drops the rejected rows and writes them to the session log file. The Secure Agent does not write the rejected rows to the reject file.
Pre SQL	Pre-SQL command to run against the target database before writing data to the target.
Post SQL	Post-SQL command to run against the target database after writing data to the target.
Update Override	An update SQL statement that updates the data in a Microsoft SQL Server target table. The update SQL statement you specify overrides the default update statements that the Secure Agent generates to update the target based on key columns. You can define an update override statement to update target tables based on both key or non-key columns. In the override statement, you must enclose all reserved words in quotation marks.
Reject file directory	The directory that stores the rejected files.
Reject filename	Name of the rejected file that is stored in the reject file directory.
Schema Name	Overrides the schema name of the target object.

Configuring an update override for the target

To override the default update SQL statement that the Secure Agent generates, you can specify an SQL statement in the **Update Override** field of the advanced target properties.

1. Next to the **Update Override** field, click **Configure**.

2. In the **Update Override SQL Editor** dialog box, enter the update SQL statement that the Secure Agent must use.
3. Click **Generate SQL** to generate an SQL query.
4. Click **Format SQL** to format the SQL query you entered.

   You can modify the generated SQL query in the SQL editor based on your requirement.
5. Click **OK**.

Microsoft SQL Server lookups in mapping

You can create lookups for objects using a Microsoft SQL Server connection. You can retrieve data from a Microsoft SQL Server lookup object based on the specified lookup condition.

When you configure a lookup in Microsoft SQL Server, you select the lookup connection and lookup object.

Note: You can't configure a Lookup transformation in a data transfer task.

The following table describes the Microsoft SQL Server lookup object properties that you can configure in a Lookup transformation:

Property	Description
Connection	Name of the lookup connection. You can select an existing connection, create a new connection, or define parameter values for the lookup connection property. If you want to overwrite the lookup connection properties at runtime, select the **Allow parameter to be overridden at run time** option. Specify the parameter file directory and name in the advanced session properties.
Source Type	Type of the Microsoft SQL Server lookup object available. Select one of the following lookup object types: - Single Object - Query - Parameter When the lookup source is large, you can use a custom query to reduce the number of columns to query.
Parameter	A parameter file where you define values that you want to update without having to edit the task. Select an existing parameter for the lookup object or click **New Parameter** to define a new parameter for the lookup object. The **Parameter** property appears only if you select parameter as the source type. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option. When the task runs, the Secure Agent uses the parameters from the file that you specify in the advanced session properties.
Lookup Object	Name of the Microsoft SQL Server lookup object.

Property	Description
Multiple Matches	The behavior when the lookup condition returns multiple matches. You can select one of the following options: - Return first row - Return last row - Return any row - Return all rows - Report error
SQL Override	The SQL statement to override the default query that creates lookup data from a Microsoft SQL Server source.

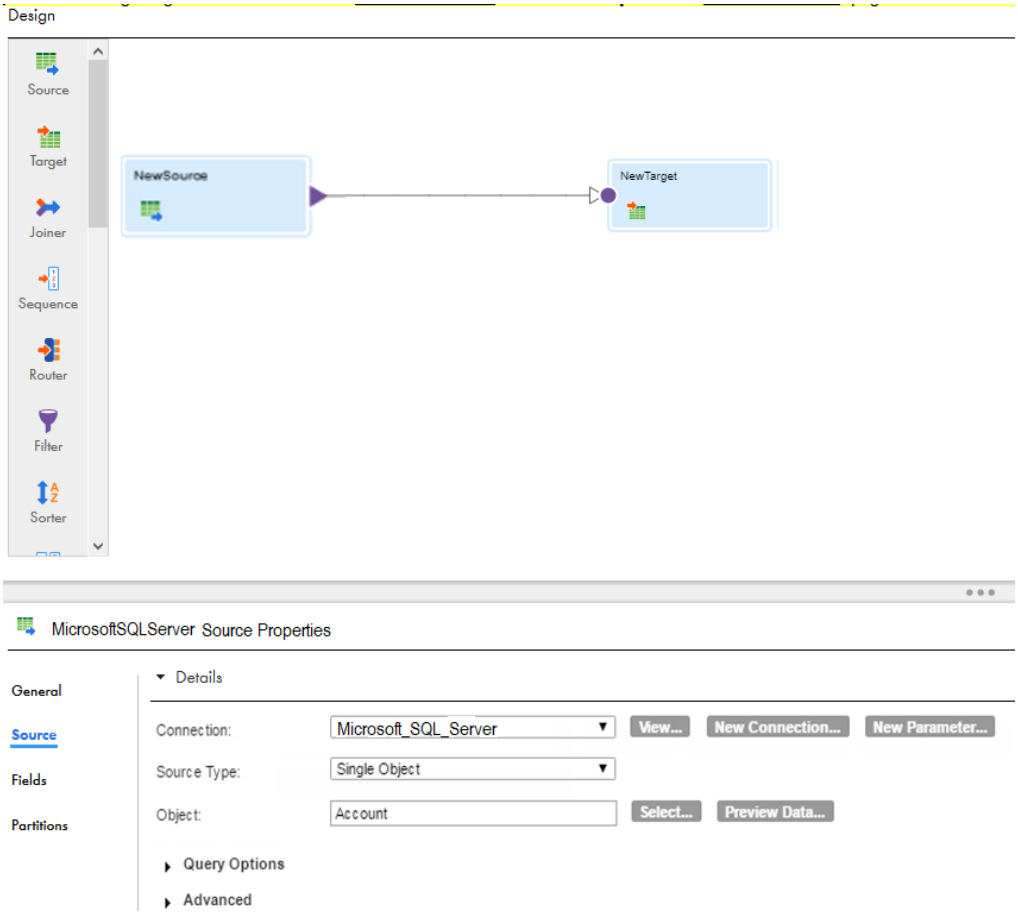
Microsoft SQL Server mapping task example

You can create a mapping task to read data from a Microsoft SQL Server source and write data to the Microsoft SQL Server target.

Perform the following task to create a mapping task:

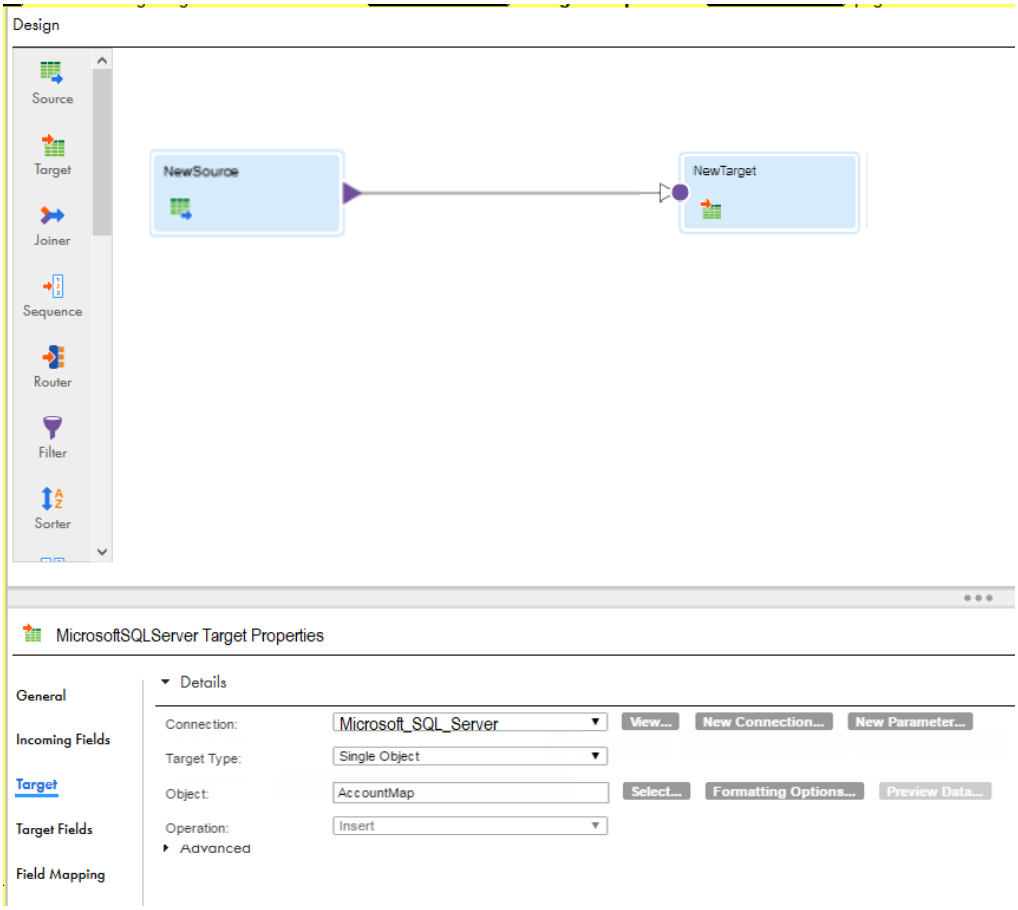
1. In Data Integration, click **New** > **Mapping** > **Create** .
2. To configure the source properties and the advanced source properties, select **Source** from the selection panel under the **Design** section.

The following image shows details of the **Source Properties** page:



3. To configure the target properties and the advanced target properties, select **Target** from the selection panel under the **Design** section.

The following image shows details of the **Target Properties** page:



4. Map the source and target.



5. Click **Save** > **Run**.

   In Monitor, you can monitor the status of the logs after you run the task.

Rules and guidelines for Microsoft SQL Server tasks

Consider the following rules and guidelines when you configure a task:

- When you use a custom or saved query in a task and the query contains * instead of a column list, NULL values are returned. You must use a column list instead of * in the query.
- When you run a mapping task to create a target, the target is successfully created at runtime. If you drop the created target table and run the task again, the task fails.

- When you run a task to update or upsert data in bulk to an Azure SQL Database target, the Secure Agent logs collective statistics of both the update and the insert operations. In normal mode, the Secure Agent logs the statistics of the update and insert operations separately.
- When you use a lookup query or lookup SQL override query in a task and the query contains dynamic column names as ?column_name?, NULL values are returned. To resolve the dynamic column names, provide the query in the **Lookup SQL Override** field and disable the lookup cache.
- Consider the following rules and guidelines when you add objects with a hyphen in the column names in the Source and Target transformations:
  - When the source column name that you read has hyphens and you use the **Create New at Runtime** option to create a target, the hyphens in the table name are converted to underscores in the target column.
  - When you add a new column to the object in an existing mapping and the column name has a hyphen, you cannot manually add the column in the mapping.
  - If you parameterize the target object, the manual field map option in the **Field Mapping** tab is not applicable. You must select Automatic, Completely Parameterized, or Partially Parameterized as the field map options for the mapping to work successfully.

SQL ELT optimization

When you read data from a Microsoft SQL Server source, transform the data, and write the data to a target, you can configure SQL ELT optimization to push the transformation logic to the source or target database system. If the source and target databases are the same, you can configure full SQL ELT optimization for improved performance.

When the Secure Agent applies SQL ELT optimization, it pushes transformation logic to a database. The Secure Agent translates the transformation logic into SQL queries and sends the SQL queries to the database. The database runs the SQL queries to process the transformations.

SQL ELT optimization improves mapping performance when the database can process the transformation logic faster than the Secure Agent. The Secure Agent also reads less data from the database.

The amount of transformation logic that the Secure Agent pushes to the database depends on the database, the transformation logic, and the mapping task. The Secure Agent processes all transformation logic that it cannot push to a database.

When you configure SQL ELT optimization for the mapping, the Secure Agent analyzes the optimized mapping from the source to the target or until it reaches a downstream transformation that it cannot push to the source database. The Secure Agent generates and executes a SELECT statement for each source that has transformation logic pushed down. Then, it reads the results of this SQL query and processes the remaining transformations in the mapping.

You cannot configure SQL ELT optimization for a mapping in advanced mode.

The Secure Agent can push the following transformation logic to a Microsoft SQL Server source or target:

Transformations	Supported SQL ELT optimization Type
Aggregator	Source, Full
Expression	Source, Full

Transformations	Supported SQL ELT optimization Type
Filter	Source, Full
Joiner	Source, Full
Sorter	Source, Full
Union	Source, Full
Router	Full

Full SQL ELT optimization

When the Secure Agent applies full SQL ELT optimization, it pushes all the transformation logic in the mapping to the target database. You can configure full SQL ELT optimization in the **SQL ELT Optimization** section.

Full SQL ELT optimization is ideal when the source and target are in the same connections. For example, if a mapping contains a Microsoft SQL Server source and a Microsoft SQL Server target, configure full SQL ELT optimization to push all the transformation logic for processing from a Microsoft SQL Server source database to a Microsoft SQL Server target database.

Source SQL ELT optimization

When the Secure Agent applies source SQL ELT optimization, it analyzes the mapping from source to target or until it reaches a downstream transformation it cannot push to the source database.

The Secure Agent generates and executes a SELECT statement based on the transformation logic for each transformation it can push to the database. Then, it reads the results of this SQL query and processes the remaining transformations.

You can configure a mapping to use source SQL ELT optimization if the source and target reside in different databases. For example, if a mapping contains a Microsoft SQL Server source and an Oracle target, you can configure source SQL ELT optimization to push some transformation logic for processing to the Microsoft SQL Server source.

SQL ELT optimization functions

When you use SQL ELT optimization, the Secure Agent converts the expression in the transformation by determining equivalent functions in the database. If there is no equivalent function in the database, the Secure Agent processes the transformation logic.

The following table summarizes the availability of SQL ELT optimization functions in Microsoft SQL Server:

Functions	SQL ELT optimization Type
ABS()	Source, Full
ADD_TO_DATE()	Full

Functions	SQL ELT optimization Type
ASCII() [1]	Full
AVG()	Source, Full
CEIL() [1]	Full
CHR()	Full
CONCAT() [1]	Full
COS()	Source, Full
COSH()	Full
COUNT()	Source, Full
DATE_COMPARE()	Source, Full
DATE_DIFF()[2]	Full
DECODE()	Source, Full
EXP()	Source, Full
FIRST()	Full
FLOOR() [1]	Full
GET_DATE_PART()	Full
IIF()	Source, Full
IN()	Source, Full
INSTR()	Full
IS_DATE()[2]	Full
IS_NUMBER()[2]	Full
ISNULL()	Source, Full
LAST_DAY()[2]	Full
LENGTH() [1]	Full
LN()	Full
LOG()	Full
LOWER()	Source, Full
LPAD()[2]	Full

Functions	SQL ELT optimization Type
LTRIM() [1]	Full
MAX()	Source, Full
MIN()	Source, Full
MD5() [2]	Source, Full
MOD() [1]	Full
POWER()	Source, Full
REPLACECHR()[2]	Full
REPLACESTR()[2]	Full
ROUND(NUMBER)	Full
RTRIM() [1]	Full
SIGN()	Full
SIN()	Source, Full
SINH()	Full
SOUNDEX()	Full
SQRT()	Source, Full
STDDEV()	Full
SUBSTR()	Full
SUM()	Source, Full
SYSTIMESTAMP() [1]	Full
TAN()	Source, Full
TANH()	Full
TO_BIGINT	Full
TO_CHAR(DATE) [1]	Full
TO_CHAR(NUMBER) [1]	Full
TO_DATE()	Full
TO_DECIMAL()	Full
TO_FLOAT()	Full

Functions	SQL ELT optimization Type
TO_INTEGER()	Full
TO_NUMBER()	Full
TRUNC(NUMBER)	Full
UPPER()	Source, Full
VARIANCE()	Full
[1]Applies also in Expression transformations for mappings enabled with source SQL ELT optimization. [2]Applies only in an Expression transformation.	

SQL ELT optimization variables

When you use SQL ELT optimization, the Secure Agent converts the expression in the transformation by determining equivalent variables in the database. If there is no equivalent variable in the database, the Secure Agent processes the transformation logic.

The following table summarizes the availability of SQL ELT optimization variables in Microsoft SQL Server:

Variables	SQL ELT optimization Type
SESSSTARTTIME	Full
SYSDATE	Full

Configuring SQL ELT optimization

To optimize a mapping, add the mapping to a task, and then configure SQL ELT optimization in the mapping task.

1. In the **Runtime Options** tab of the Mapping task, navigate to the **SQL ELT Optimization** section.
2. From the **SQL ELT Optimization** list, select the required type of SQL ELT optimization.

Cross-schema SQL ELT optimization

You can use cross-schema SQL ELT optimization for a mapping task to read from or write data to Microsoft SQL Server objects associated with different schemas within the same database.

To use cross-schema SQL ELT optimization, create two Microsoft SQL Server connections and specify the schema in each connection. Ensure that the schema in the source connection is different from the schema in the target connection, but both the schemas must belong to the same database. When you configure SQL ELT optimization for the mapping task, enable cross-schema SQL ELT optimization in the advanced session properties. By default, the check box is selected.

Configuring cross-schema SQL ELT optimization

Consider the following steps to configure cross-schema SQL ELT optimization for a Microsoft SQL Server mapping task:

1. Create the following two Microsoft SQL Server connections, each defined with a different schema:
   a. Create a `sqlsv_1` Microsoft SQL Server connection and specify `CQA_SCHEMA1` schema in the connection properties.
   b. Create a `sqlsv_2` Microsoft SQL Server connection and specify `CQA_SCHEMA2` schema in the connection properties.
2. Create a Microsoft SQL Server mapping, `m_sqlsv_pdo_acrossSchema`. Perform the following tasks:
   a. Add a Source transformation and include a Microsoft SQL Server source object and connection `sqlsv1` to read data using `CQA_SCHEMA1`.
   b. Add a Target transformation and include a Microsoft SQL Server target object and connection `sqlsv2` to write data using `CQA_SCHEMA2`.
3. Create a Microsoft SQL Server mapping task, and perform the following tasks:
   a. Select the configured Microsoft SQL Server mapping, `m_sqlsv_pdo_acrossSchema`.
   b. On the **Runtime Options** tab, in the **SQL ELT Optimization** section, set the pSQL ELT optimization value to **Full**.
   c. In the **Advanced Session Properties** section, select the **Enable cross-schema SQL ELT optimization** check box.
   d. Save the task and click **Finish**.

   When you run the mapping task, the Secure Agent reads data from the Microsoft SQL Server source object associated with the `CQA_SCHEMA1` schema and writes data to the Microsoft SQL Server target object associated with `CQA_SCHEMA2` schema.

Rules and guidelines for SQL ELT optimization

Consider the following rules and guidelines when you configure SQL ELT optimization for a Microsoft SQL Server mapping:

- When you push transformation logic to the database, ensure that the database has enough resources to process the queries faster. Otherwise, there could be a performance degradation.
- If the following transformation or mapping conditions is true, the Secure Agent processes a logic instead of pushing it to the database:
  - The transformation contains a variable port.
  - The transformation is not a Sorter transformation, Union transformation, or target in a mapping.
  - The transformation downstream from a Sorter or Union transformation, or contains a distinct sort.
  - A configured mapping task to override the default values of input or output ports.
  - The database does not have an equivalent operator, variable, or function to use in an expression in the transformation.
  - The mapping contains too many branches. The Secure Agent can't generate an SQL query for a mapping that contains more than 64 two-way branches, 43 three-way branches, or 32 four-way branches. If the number of branches exceeds these limitations, the Integration Service processes the downstream transformations.
  - A mapping task to log row errors.

Consider the following rules and guidelines when you configure full SQL ELT optimization for a Microsoft SQL Server mapping:

- You can't push the LTRIM(), RTRIM(), or MOD() function that contains more than one argument.
- To push the MOD() function to the Microsoft SQL Server database, the argument that you pass must be of the Integer date type.
- When you push the INSTR() function, you can only define string, search_value, and start arguments.

Consider the following rules and guidelines when you use an Expression transformation in a Microsoft SQL Server mapping enabled with full SQL ELT optimization:

- When you push the IS_DATE() function that contains the Text or Ntext data type, the IS_DATE() function returns 0.
- When you push the IS_DATE() and IS_NUMBER() functions and the values in the argument contain NULL, the functions return 0.
- You can't get a case-sensitive return value for the REPLACECHR() or REPLACESTR() functions.
- When you push the MD5() function that contains the nchar data type, the function returns a different value for the nchar data type as compared to a mapping that runs without SQL ELT optimization.
- To get the same return value for the nchar and char data types when you push the MD5() function, enter the `ConvertToVarcharForMD5InPDO=Yes` property in the mapping task.
- When you push the LPAD() function that contains the second_string argument, the function truncates the second string from left to right.
  For example, for the LPAD('Infa',9,'RELATIONAL CONNECTIVITY') expression, the function returns the following value: IVITYInfa
- When you push the DATE_DIFF() function that contains the date1 and date2 arguments, the function returns the following different values as compared to a mapping that runs without SQL ELT optimization:
  - The function returns a negative number when the value of date1 is later than the value of date2.
  - The function returns a positive number when the value of date1 is earlier than the value of date2.

CHAPTER 5

Replication tasks with Microsoft SQL Server

You can replicate Microsoft SQL Server data to a target using the replication task. You might replicate data to back up the data or perform offline reporting. You can replicate data in Microsoft SQL Server objects to databases or flat files.

A replication task can replicate data from one or more Microsoft SQL Server objects. When you configure the task, you can replicate all available objects through the selected connection, or you can select objects for replication by including or excluding a set of objects.

You can also exclude rows and columns from the replication task. Associate a schedule with a replication task to specify when and how often the task runs.

Note: The length of source tables and column names can contain at most 64 characters.

The maximum number of characters that a replication task can write to each row in a Microsoft SQL Server 2000 target is 8060. If a replication task tries to write more than the maximum amount of characters to a row, the task fails with the following error: `WRT_8229 Database errors occurred: FnName: Execute -- [Microsoft][ODBC SQL Server Driver][SQL Server]Cannot create a row of size <row size> which is greater than the allowable maximum of 8060. FnName: Execute -- [Microsoft][ODBC SQL Server Driver][SQL Server]The statement has been terminated.`

Microsoft SQL Server sources in replication tasks

When you replicate a source object to a database, the replication task replicates the data to a database table with the same name as the source object.

The following table describes the Microsoft SQL Server source properties in a replication task:

Property	Description
Source Connection	Name of the active source connection. Select a source connection or create a new connection.
Objects to Replicate	Source objects that you can replicate. You can choose from the following options to replicate the objects: - All Objects - Include Objects - Exclude Objects
If an error occurs while processing an object	Determines the execution or termination of the process when an error occurs.

Microsoft SQL Server targets in replication tasks

When you replicate data to a database target, the replication task truncates the table name in the target database when the source object name exceeds 64 characters.

The following table describes the target properties in a replication task:

Property	Description
Connection	Name of the target connection. Select a target connection or create a connection.
Target Prefix	String that prefixes the source object names to create names for the target objects in the target. For more information, see “Table Name Truncation” in the Tasks help.
Enable Target Bulk Load	Uses Microsoft SQL Server bulk API. When you replicate data from Salesforce to Microsoft SQL Server in bulk mode, the task fails. This issue occurs if you change the datatype, precision, or scale of a Salesforce field and re-run the task.
Load Type	Replicates full load of data from Microsoft SQL Server to other applications.
Delete Options	This property is not applicable for Microsoft SQL Server Connector.

You can also configure the commit size, which defines the number of rows to commit. If you do not specify a value, the Secure Agent uses the default value. The default value is 10000.

CHAPTER 6

Data type reference

Data Integration uses the following data types in mappings, synchronization tasks, replication tasks, and mapping tasks with Microsoft SQL Server:

Microsoft SQL Server native data types

Microsoft SQL Server data types appear in the source and target transformations when you choose to edit metadata for the fields.

Transformation data types

Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Secure Agent uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

When Data Integration reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When Data Integration writes to a target, it converts the transformation data types to the comparable native data types.

Microsoft SQL Server and transformation data types

The following table lists the Microsoft SQL Server data types that Data Integration supports and the corresponding transformation data types:

Microsoft SQL Server Data Type	Description	Transformation Data Type	Description
Bigint	-9223372036854775808 to 9223372036854775807	Bigint	-9223372036854775808 to 9223372036854775807 Precision 19, Scale 0
Binary	1 to 8000 characters	Binary	1 to 104857600 bytes Precision 1 to 8000
Bit	0 to 1	Integer	0 to 1 Precision 10, Scale 0
Char	1 to 8000 characters	String	1 to 104857600 characters
Date	0001-01-01 to 9999-12-31	Date/Time	0001-01-01 to 9999-12-31 Precision 29, Scale 9

Microsoft SQL Server Data Type	Description	Transformation Data Type	Description
Datetime	**Date range**: January 1, 1753 to December 31, 9999 **Time range**: 00:00:00 to 23:59:59.997 Precision 23, scale 3	Date/Time	**Date range**: 0001-01-01 to 9999-12-31 **Time range**: 00:00:00 to 23:59:59.9999999 Precision 29, Scale 9
Datetime2	**Date range**: 0001-01-01 to 9999-12-31 **Time range**: 00:00:00 to 23:59:59.9999999 Precision 27, scale 7	Date/Time	**Date range**: 0001-01-01 to 9999-12-31 **Time range**: 00:00:00 to 23:59:59.9999999 Precision 29, Scale 9
Decimal(P,S)	Precision 1 to 28, Scale 0 to 28	Decimal	Precision 1 to 28, Scale 0 to 28
Float	-1.79E+308 to 1.79E+308	Double	Precision 15, Scale 0
Int	-2147483648 to 2147483647	Integer	-2147483648 to 2147483647 Precision 10, Scale 0
Money	-922337203685477.5808 to 922337203685477.5807	Decimal	-922337203685477.5808 to 922337203685477.5807 Precision 19, Scale 4
Nchar	1 to 4000 characters	String	1 to 104857600 characters
Ntext	1 to 1073741823 bytes	Text	1 to 104857600 characters Default is 32000
Numeric(P,S)	Precision 1 to 28, Scale 0 to 28	Decimal	Precision 1 to 28, Scale 0 to 28
Nvarchar	1 to 4000 characters	String	1 to 104857600 characters
Real	-3.40E+38 to 3.40E+38	Decimal	Precision 7, Scale 0
Smalldatetime	**Date range**: 1900-01-01 to 2079-06-06 **Time range**: 00:00:00 to 23:59:59	Date/Time	**Date range**: 0001-01-01 to 9999-12-31 **Time range**: 00:00:00 to 23:59:59.9999999 Precision 29, Scale 9
Smallint	-32768 to 32767	Integer	-32768 to 32767 Precision 10, Scale 0
Smallmoney	-214748.3648 to 214748.3647	Decimal	-214748.3648 to 214748.3647 Precision 10, Scale 4
Text	1 to 2147483647 bytes	Text	1 to 104857600 characters Default is 32000

Microsoft SQL Server Data Type	Description	Transformation Data Type	Description
Time	00:00:00.0000000 to 23:59:59.9999999	Date/Time	**Date range**: 0001-01-01 to 9999-12-31 **Time range**: 00:00:00 to 23:59:59.9999999 Precision 29, Scale 9
Tinyint	0 to 255	Integer	0 to 255 Precision 10, Scale 0
Varbinary	1 to 8000 characters	Binary	1 to 104857600 bytes Precision 1 to 8000
Varchar	1 to 8000 characters	String	1 to 104857600 characters

INDEX

C

Cloud Application Integration community
  URL 5
Cloud Developer community
  URL 5
connections
  Microsoft SQL Server 10

D

Data Integration community
  URL 5
data type reference
  overview 42

E

environment variables
  Kerberos authentication 9

F

functions
  SQL ELT optimization 34

I

Informatica Global Customer Support
  contact information 6
Informatica Intelligent Cloud Services
  web site 5

K

Kerberos authentication
  Microsoft SQL Server 8
key range partitioning
  configuration 26

M

maintenance outages 6
mapping
  mapping task 23
mapping task
  example 30
Microsoft SQL Server
  assets 7
  connection properties 10
Microsoft SQL Server *(continued)*
  data types 42
Microsoft SQL Server advanced properties
  Synchronization task 19
Microsoft SQL Server connector
  rules and guidelines 38
Microsoft SQL Server Connector
  overview 7
  administration 16
Microsoft SQL Server lookup
  Synchronization task 20
Microsoft SQL Server sources
  mapping 23, 29
  Synchronization task 18
Microsoft SQL Server targets
  mapping 27
  Synchronization task 19

P

pass-through partitioning 26

R

replication task
  overview 40, 41

S

SQL ELT optimization
  functions 34
  variables 37
SQL ELT optimization method
  full SQL ELT optimization 34
  source SQL ELT optimization 34
status
  Informatica Intelligent Cloud Services 6
Synchronization task
  example 20
system status 6

T

trust site
  description 6

U

upgrade notifications 6

V

variable
SQL ELT optimization 37

W

web site 5