Informatica®

Informatica® Cloud Data Integration

Teradata Connector

Informatica Cloud Data Integration Teradata Connector
July 2024

© Copyright Informatica LLC 2015, 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 Teradata Connector 7

Teradata Connector assets 7

Chapter 2: Connections for Teradata 8

Prerequisites 8

- Teradata Parallel Transporter Utilities 8
- Configuring Kerberos authentication 8
- Setting environment variables 9

Connect to Teradata 10

- Before you begin 10
- Connection details 10
- Authentication types 11
- Advanced settings 12

Database privileges 12

Chapter 3: Teradata sources and targets 14

Teradata Sources 14

- Tracing Levels 14
- Spool Modes 15
- SQL Override Query 15

Teradata Targets 15

- System Operators 16
- Row-level Processing 17
- Working with Log, Error, and Work Tables 18
- Drop Log, Error, and Work Tables 19

Chapter 4: Mappings and Mapping Tasks with Teradata 20

Teradata Objects in Mappings 20

- Teradata Sources in Mappings 20

Partitioning for Teradata Sources. 22
Teradata Targets in Mappings. 22
Partitioning for Teradata Targets. 26
Mapping Example. 26
Calling a stored procedure. 27

Chapter 5: Teradata ODBC SQL ELT optimization. 28

Supported transformations. 28
Supported functions. 29
Supported variables. 31
Configuring the Teradata ODBC driver. 31
Configuring Teradata ODBC driver on Windows. 31
Configuring the Teradata ODBC driver on Linux. 31
Configuring a Teradata ODBC connection. 32
Configuring SQL ELT optimization. 32
Configuring cross-schema SQL ELT optimization for a Teradata ODBC mapping. 33
Enabling SQL ELT optimization for Expression and Aggregator transformations. 34
Rules and guidelines for SQL ELT optimization. 34

Chapter 6: Data Type Reference. 35

Teradata Data Types and Transformation Data Types. 35

Index. 37

Preface

Use *Teradata Connector* to learn how to read from or write to Teradata by using Cloud Data Integration. Learn to create a Teradata connection, develop mappings, and run mapping tasks in Cloud Data Integration. You can also learn how to configure SQL ELT optimization to a Teradata database using a Teradata ODBC connection.

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 Teradata Connector

Teradata Connector integrates Data Integration and Teradata Parallel Transporter Application Programming Interface (Teradata PT API) for data extraction and loading.

Teradata PT is a load and unload utility that extracts, transforms, and loads data from multiple sources in parallel. Teradata PT API provides high-speed, scalable, parallel data extraction, and bulk data load and update to the Teradata database.

To use Teradata Connector, create a task with a Teradata source or target. Use a Teradata connection in the task to connect to the Teradata tables that you want to export or load in a task. The Secure Agent uses the Teradata PT API infrastructure to connect to Teradata.

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

The Secure Agent loads data into Teradata using one of the following system operators that you specify in the mapping:

Load

Bulk loads data into an empty Teradata database table.

Update

Performs update, insert, upsert, and delete operations against Teradata database tables.

Stream

Performs update, insert, upsert, and delete operations against Teradata database tables in near real-time mode.

Teradata Connector assets

Create assets in Data Integration to integrate data using Teradata Connector.

When you use Teradata Connector, you can include the following Data Integration assets:

- Mapping
- Mapping task

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

CHAPTER 2

Connections for Teradata

Create a Teradata connection to read data from or write data to Teradata.

You can use a Teradata connection to specify sources and targets in mappings and mapping tasks.

Prerequisites

Before you can use Teradata Connector, ensure that you meet the prerequisite tasks.

Perform the following prerequisite tasks:

1. Install the Teradata Parallel Transporter utilities and set the environment variables.
2. Ensure that you have access to the Secure Agent directory that contains the success and error files. This directory path must be the same on each Secure Agent machine in the runtime environment. The Teradata JDBC driver is packaged with the Secure Agent. When you install the Secure Agent, the JDBC driver is installed and the JDBC jars are copied to the Secure Agent machine.
3. Configure the authentication prerequisites to connect to a Teradata database. You can configure Native, LDAP, or KRB5 authentication. Keep the authentication details handy based on the authentication type that you want to use.

Teradata Parallel Transporter Utilities

Before you use Teradata Connector, install the Teradata Parallel Transporter utilities on the Secure Agent machine.

You must install the following Teradata Parallel Transporter utilities:

- Teradata Parallel Transporter Base
- Teradata Parallel Transporter Stream Operator
- Teradata CLIv2
- Teradata Generic Security Services
- Shared ICU Libraries for Teradata

Configuring Kerberos authentication

You can configure Kerberos for a Secure Agent only on a Linux machine. Configure the Secure Agent to use Kerberos to authenticate when you connect to a Teradata database.

Before your configure Kerberos authentication, ensure that you have the required Kerberos configuration files, such as krb5.conf and IICSTPT.keytab. The Secure Agent uses the IICSTPT.keytab file to generate Kerberos TGT cache entries at runtime.

Create the Kerberos configuration files

The Secure Agent requires the keytab files to authenticate users to connect to Teradata.

To generate the keytab file, perform the following steps:

1. Edit the `krb5.conf` and set `default_tgs_enctypes = arcfour-hmac-md5`.
2. Add the encryption configuration property in the `krb5.conf` file in the `[libdefaults]` section.
   For example, set the following entries for the klist ouput in the keytab file:

   ```
   [devbld@invlxrhadpdev05
   jdbcConnectionSample]$ klist -ekt IICSTPT.keytab
   Keytab name: FILE:IICSTPT.keytab
   KVNO Timestamp Principal
   1 12/11/2018 15:02:06 teradbqa_mit@INFAQAKERB
   (arcfour-hmac)
   ```

   The Secure Agent uses the keytab file information that you specify and creates the Kerberos TGT cache in the `$PMTempDir` directory at runtime. The Kerberos TGT cache is active during the length of the session and ends when the session completes.

Add the Kerberos configuration files to the Kerberos artifacts directory

Add the `krb5.conf` and `IICSTPT.keytab` files to a directory on the Secure Agent machine.

When you select KRB5 as the authentication type in the Teradata connection, specify the directory that contains these files in the **Kerberos Artifacts Directory** field.

Setting environment variables

You must configure Java and Teradata environment variables before you can use Teradata Connector.

The following table describes the environment variables you must set on UNIX:

Environment Variable	Value
THREADONOFF	On UNIX and Linux, set the `THREADONOFF` environment variable to 1 to enable multithreading support for Teradata processes.
NLSPATH	Set the NLSPATH variable to the location of the `opermsgs.cat` file. For example, `/opt/teradata/client/15.10/msg/%N`

Also, set the shared library environment variable based on the operating system.

The following table describes the shared library variables for each operating system:

Operating System	Value
Windows	`PATH`
Linux	`LD_LIBRARY_PATH`

For example, use the following syntax for Linux:

- Using a Bourne shell:

```
export LD_LIBRARY_PATH="/opt/teradata/client/15.10/lib64:${LD_LIBRARY_PATH}"
```

- Using a C shell:

```
LD_LIBRARY_PATH="/opt/teradata/client/15.10/lib64:${LD_LIBRARY_PATH}
```

After you set the environment variables, restart the Secure Agent.

Connect to Teradata

Let's configure the Teradata connection properties to connect to Teradata.

Before you begin

Before you get started, be sure to complete the prerequisites.

Check out "Prerequisites" on page 8 to learn more about the authentication prerequisites and tasks that you must perform.

Connection details

The following table describes the basic 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	Teradata
Runtime Environment	The name of the runtime environment where you want to run tasks. Specify a Secure Agent.
TDPID	The name or IP address of the Teradata database machine.
Database Name	The Teradata database name. If you do not enter a database name, Teradata PT API uses the default login database name.
Code Page	Code page associated with the Teradata database. Select one of the following code pages: - MS Windows Latin 1. Select for ISO 8859-1 Western European data. - UTF-8. Select for Unicode and non-Unicode data. When you run a task that extracts data from a Teradata source, the code page of the Teradata PT API connection must be the same as the code page of the Teradata source.

Authentication types

You can configure Native, LDAP, and KRB5 authentication types to connect to the Teradata database.

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

Native authentication

To configure native authentication, you need the user name and password from your Teradata account.

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

Property	Description
Username	User name with the appropriate read and write database permissions to access the Teradata database.
Password	Password for the database user name.

LDAP authentication

To configure LDAP authentication, you need to authenticate user credentials against the external LDAP directory service.

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

Property	Description
Username	User name with the appropriate read and write database permissions to access the LDAP database.
Password	Password for the LDAP database user name.

KRB5 authentication

To configure KRB5 authentication, you must configure the Secure Agent hosted on the Linux machine and ensure that the Kerberos artifacts directory contains the Kerberos configuration files.

The following table describes the basic connection properties for KRB5 authentication:

Property	Description
Username	The Kerberos user name that authenticates to the Teradata database.

The following table describes the advanced connection properties for KRB5 authentication:

Property	Description
Kerberos Artifacts Directory	Directory that contains the Kerberos configuration files. This directory contains the following configuration files: - krb5.conf - IICSTPT.keytab

Advanced settings

The following table describes the advanced connection properties:

Property	Description
Tenacity	Amount of time, in hours, that the Teradata PT API continues attempting to log in again when the maximum number of operations runs on the Teradata database. Specify a positive integer. Default is 4.
Max Sessions	Maximum number of sessions allowed for the Teradata PT API to establish connection with the Teradata database. Specify a positive, non-zero integer. Default is 4.
Min Sessions	Minimum number of Teradata PT API sessions required for the Teradata PT API job to continue. Specify a positive integer between 1 and the value specified in the **Max Sessions** field. Default is 1.
Sleep	Amount of time, in minutes, that the Teradata PT API waits before it attempts to log in again when the maximum number of operations run on the Teradata database. Specify a positive, non-zero integer. Default is 6.
Data Encryption	Enables full security encryption of SQL requests, responses, and data. Default is disabled.
Block Size	Maximum block size, in bytes. Teradata PT API uses this property to read the data block size from source through the Export operator. Maximum is 16775168 bytes for Teradata database version 16.20 and later. If the Teradata database version is earlier than 16.20, Teradata scales down the block size from 16775168 bytes to the maximum allowed value. The block size 16775168 is not allowed in the Spool mode. For more information, see the Teradata logs and verify the Teradata documentation of the same version.
Metadata Advanced Connection Properties	The optional properties for the JDBC driver to fetch the metadata from Teradata. For example, tmode=ANSI.
Enable Metadata Quotification	Determines if a Teradata connection reads reserved words used in table or column names from the Teradata database. Select the **Enable Metadata Quotification** checkbox for the Secure Agent to read reserved words from Teradata.

Database privileges

Before you use Teradata Connector, select privileges on specific Data Dictionary tables.

Verify that you have the following database privileges:

- Select privileges on DBC.Tables, DBC.Columns, DBC.UDTInfo, and DBC.Databases.
  For information about select privileges, see the Teradata JDBC Driver documentation.

CHAPTER 3

Teradata sources and targets

You can add a Source transformation to extract data from a source. You can add a Target transformation to write data to a target.

When you add a Source transformation to a mapping, you define the source connection, source objects, and source properties related to the Teradata connection type.

When you add a Target transformation to a mapping, you define the target connection, target objects, and target properties related to the Teradata connection type.

Teradata Sources

You can use a Teradata object as a source in a mapping or mapping task. When you configure a Teradata task, you define properties that determine how the Secure Agent extracts data from Teradata sources.

When you use Teradata source objects, you can select a single Teradata source object or multiple source objects. When you configure the advanced source properties, you configure properties specific to Teradata.

You can configure tracing levels, spool modes, and query band expressions when you use Teradata sources to read data from Teradata.

Tracing Levels

You can specify the tracing level for the driver, or for activities related to the Teradata PT infrastructure. Tracing determines the type of diagnostic messages that the system operator sends to the log file. You can specify the absolute path with the trace file name in the advanced source properties.

Tracing helps in providing detailed information in the log file that aids in problem tracking and diagnosis.

You can specify the following tracing functions at the driver or Teradata PT infrastructure level for Teradata PT API:

TD_OFF

Teradata PT API disables the tracing. Default is TD_OFF.

TD_OPER

Teradata PT API enables tracing for driver-specific activities for Teradata.

TD_OPER_CLI

Teradata PT API enables tracing for activites involving CLIv2.

TD_OPER_NOTIFY

Teradata PT API enables tracing for activities involving the Notify feature.

TD_OPER_OPCOMMON

Teradata PT API enables tracing for activities involving the operator common library.

TD_OPER_ALL

Teradata PT API enables all driver-level tracing.

Spool Modes

You can configure a task so that Teradata PT API uses one of the spool modes to extract data from Teradata. By default, Teradata PT API spools data while extracting data from Teradata. For information about the spool modes and their uses, see the Teradata documentation.

You can configure the following source properties for Spool Mode that Teradata PT API uses to extract data from Teradata:

Spool

Teradata PT API spools data while extracting data from Teradata. Data is stored in a buffer and then extracted.

NoSpool

Teradata PT API does not spool data while extracting data from Teradata. The NoSpool mode extracts data quickly without reading the data into a spool file before extracting data. If the database does not support the NoSpool option, Teradata PT API uses the Spool option.

NoSpoolOnly

Teradata PT API does not spool while extracting data from Teradata. If the database does not support NoSpool, the task fails with an error.

SQL Override Query

When you configure a mapping, you can specify an SQL statement to override the default query used to read data from the Teradata source.

Specify the SQL override query in the source properties of the mapping. In the SQL override query, you can specify the columns that you want to use from the source database.

For example, use the following SQL override query to extract employee records, matching rows by employee location, from the Teradata source:

```
select EMPID, EMPNAME, AGE, DOB, DEPTID, LOCATION, DESIGNATION, SALARY from employee where CITY='San Jose'
```

When you run the mapping, the Secure Agent extracts specific employee records belonging to San Jose based on the where clause defined in the SQL override query.

You must ensure that the number of columns, order of columns, and the data type you specify in the query matches with the Teradata source object.

Teradata Targets

You can use a Teradata object as a single target in a mapping or mapping task.

You can insert, update, upsert, and delete data from Teradata targets. When you configure the advanced target properties, you configure properties specific to Teradata.

System Operators

Teradata PT API uses the Load, Update, and Stream system operators to write data to Teradata.

Specify the type of system operator your require in the target properties. When you run the task, the Secure Agent uses Teradata PT API to connect to Teradata to load, update, or stream data by using the Teradata PT API Load, Update, or Stream system operators.

Load System Operator

The Load system operator loads a large volume of data at high speed into an empty Teradata table. You can use the Load operator to load initial data to Teradata tables as it inserts data into individual rows of a target table.

Before you load data to Teradata tables, you must ensure that the target table is empty.

Update System Operator

You can use the Update system operator to perform insert, update, upsert, and delete operations against Teradata database tables. You can load data to empty or existing Teradata tables.

You can perform update, upsert, and delete operations after you define a primary key in the target table. The Secure Agent does not retain the primary key information of a Teradata table during metadata import. You must therefore select the target columns in the **Update Columns** field in the target properties. The target columns serve as a key when you run an update, upsert, or delete operation.

Stream System Operator

You can use the Stream system operator to perform high-speed parallel inserts to empty or existing Teradata tables without locking target tables.

You can perform update, upsert, and delete operations after you define a primary key in the target table. The Secure Agent does not retain the primary key information of a Teradata table during metadata import. You must therefore select the target columns in the **Update Columns** field in the target properties. The target columns serve as a key when you run an update, upsert, or delete operation.

You can maintain current and accurate data for immediate decision making. The Stream system operator uses macros to modify tables. The Stream system operator creates macros before Teradata PT API begins loading data and removes them from the database after Teradata PT API loads all rows to the target. If you do not specify a macro database, Teradata PT API stores the macros in the log database.

Error Limit for the Stream Operator

Teradata PT API terminates the Stream system operator job after it reaches the maximum number of records that can be stored in the error table.

The error limit is approximate because the Stream operator sends multiple rows of data at a time to the Teradata table. By the time Teradata PT API processes the message that indicates that the error limit has been exceeded, it might have loaded more records into the error table than the number specified in the error limit.

By default, the error limit value is unlimited. The error limit applies to each instance of the Stream system operator.

Row-level Processing

When you write data to a Teradata target, you can configure how Teradata PT API treats duplicate rows, missing rows, and extra rows.

Duplicate Rows

You can configure how Teradata PT API handles duplicate rows when it tries to insert or upsert rows in the target table.

You can select one of the following values:

- None. If Teradata PT API receives a row marked for insert or upsert that causes a duplicate row in the target table, Teradata PT API does not write the row to the error table and does not mark it as an error row in the session log.
- For Insert. If Teradata PT API receives a row marked for insert that exists in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.
- For Update. If Teradata PT API receives a row marked for update that causes a duplicate row in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.
- Both. If Teradata PT API receives a row marked for insert or upsert that causes a duplicate row in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.

Missing Rows

You can configure how Teradata PT API handles rows that do not exist in the target table when it tries to update or delete rows.

You can select one of the following values:

- None. If Teradata PT API receives a row marked for update or delete but it is missing in the target table, Teradata PT API does not write the row to the error table and does not mark it as an error row in the session log.
- For Update. If Teradata PT API receives a row marked for update but it is missing in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.
- For Delete. If Teradata PT API receives a row marked for delete but it is missing in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.
- Both. If Teradata PT API receives a row marked for update or delete but it is missing in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.

Extra Rows

You can configure how Teradata PT API marks error rows when it tries to update or delete multiple target table rows.

You can select one of the following values:

- None. If Teradata PT API receives a row marked for update or delete that affects multiple rows in the target table, Teradata PT API does not write the row to the error table and does not mark it as an error row in the session log.
- For Update. If Teradata PT API receives a row marked for update that affects multiple rows in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.
- For Delete. If Teradata PT API receives a row marked for delete that affects multiple rows in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.

- Both. If Teradata PT API receives a row marked for update or delete that affects multiple rows in the target table, Teradata PT API writes the row to the error table and marks it as an error row in the session log.

This attribute is available only for the Stream system operator.

Working with Log, Error, and Work Tables

When you run a session that extracts data from or loads data to Teradata using Teradata PT API, Teradata PT API creates the following tables:

- **Log Tables.** Stores Teradata PT API restart and log information. Teradata PT API creates one log table for each partition.
- **Error Tables.** Logs Teradata errors and rejected data when a session runs. Teradata PT API creates two error tables for each partition.
- **Work Tables.** Stores¾ data when you run a session that uses the Update system operator. Teradata PT API creates one work table for each partition.

Log Tables

Enter a log table name when you configure a task to load to Teradata. You can also choose to create the log table in a log database, a working database, or under the default database. Choose where you want to create the log table when you configure a task to load data to Teradata.

The following table describes the advanced target properties that allow you to specify the log table information:

Property	Description
Log Database	Name of the database that stores the log tables. If you do not enter a log database name in the task properties or a database name in the connection object, Teradata PT API stores the log tables under the user.
Log Table Name	Name of the log table. If you do not specify a log table name, the Secure Agent uses the name *<log_database>.INFA_LT_<number>*. The exact table name appears in the session log.

When a task fails, see the log table for more information. Before you run the task again, drop the log table or enter a different table name in the advanced target properties.

Error Tables

Teradata writes rejected data to error tables ErrorTable1 and ErrorTable2.

ErrorTable1 contains data rejected for the following reasons:

- Data conversion errors
- Constraint violations
- Access Module Processor configuration changes

ErrorTable2 contains data rejected for the following reasons:

- Unique primary index constraint violations
- Load driver job acquisition phase errors

You can enter a name for each error table when you configure a task to load data to Teradata. You can also choose to create the error tables in an error database, a working database, or under the default database. Choose where you want to create the error tables when you configure a task to load data to Teradata.

The following table describes the advanced target properties that allow you to specify error table names:

Property	Description
Error Database	Name of the database that stores the error tables. If you do not enter an error database name in the advanced target properties or a database name in the connection object, Teradata PT API stores the error tables under the user.
Error Table Name1	Name of the first error table. If you do not specify a name for the first error table, the Secure Agent uses the name *<error_database>.INFA_ET1_<number>*. The exact table name appears in the session log.
Error Table Name2	Name of the second error table. If you do not specify a name for the second error table, the Secure Agent uses the name *<error_database>.INFA_ET2_<number>*. The exact table name appears in the session log.

When a task fails, see the error tables for more information about the errors. Before you run the task again, drop the error tables or enter different table names in the advanced target properties.

Work Tables

The Update system operator uses DML statements for staging data. It creates work tables before Teradata PT API begins loading data and removes them from the database after Teradata PT API loads all rows to the target.

Enter a work table name when you configure a task to load data to Teradata. You can also choose to create the work table in the target database. Choose where you want to create the work table when you configure a task to load data to Teradata.

The following table describes the advanced target properties that allow you to specify work table information:

Property	Description
Work Table Database	Name of the database that stores the work tables created by Teradata PT API when you select the Update system operator. If you do not specify a work table database, Teradata PT API stores the work tables in the target database.
Work Table Name	Name of the work tables when you select the Update system operator. The Teradata database creates one work table for each target table. If you do not specify a work table name, the Secure Agent uses the name *<work_table_database>.INFA<number>*_WT. The exact table name appears in the session log.

Drop Log, Error, and Work Tables

If you configure a task to use a Teradata target connection, and enable the **Drop Log/Error/Work Tables** Teradata target advanced property, the Secure Agent drops existing log, error, and work tables for a session when the session starts.

CHAPTER 4

Mappings and Mapping Tasks with Teradata

Use the Data Integration Mapping Designer to create a mapping. In advanced mode, the Mapping Designer updates the mapping canvas to include transformations and functions that enable advanced functionality.

Teradata Objects in Mappings

When you create a mapping, you can configure a Source or Target transformation to represent a Teradata object. After you configure a mapping, deploy the mapping in a mapping task.

Teradata Sources in Mappings

In a mapping, you can configure a Source transformation to represent a single Teradata source or multiple Teradata sources.

The following table describes the Teradata source properties that you can configure in a Source transformation:

Property	Description
Connection	Select the source connection, or click **New Connection** to create one. Alternatively, you can define a connection parameter in the mapping and enter a specific connection in each mapping task that is associated with the mapping.
Source type	Type of the source object. Select one of the following types: - Single Object. Select to specify a single Teradata object. - Multiple Objects. Select to specify multiple Teradata objects. - Parameter. Select to specify a parameter name. You can configure the source object in a mapping task associated with a mapping that uses this source transformation.
Object	Name of the source object. Select the source object for a single source. When you select the multiple source objects option, you can add multiple related source objects and configure relationships between them. You can use existing relationships or custom relationships to join multiple source objects. When you create a custom relationship for Teradata objects, you can select the type of join and the source fields to use. If you select Parameter, create or select the parameter for the source object.

The following table describes the Teradata source advanced properties that you can configure in a Source transformation:

Advanced Property	Description
Driver Tracing Level	Determines Teradata PT API tracing at the driver level: - TD_OFF. Teradata PT API disables tracing. - TD_OPER. Teradata PT API enables tracing for driver-specific activities for Teradata. - TD_OPER_CLI. Teradata PT API enables tracing for activites involving CLIv2. - TD_OPER_NOTIFY. Teradata PT API enables tracing for activities involving the Notify feature. - TD_OPER_OPCOMMON. Teradata PT API enables tracing for activities involving the operator common library. - TD_OPER_ALL. Teradata PT API enables all driver-level tracing. Default is TD_OFF.
Infrastructure Tracing Level	Determines Teradata PT API tracing at the infrastructure level: - TD_OFF. Teradata PT API disables tracing. - TD_OPER. Teradata PT API enables tracing for driver-specific activities for Teradata. - TD_OPER_CLI. Teradata PT API enables tracing for activites involving CLIv2. - TD_OPER_NOTIFY. Teradata PT API enables tracing for activities involving the Notify feature. - TD_OPER_OPCOMMON. Teradata PT API enables tracing for activities involving the operator common library. - TD_OPER_ALL. Teradata PT API enables all driver-level tracing. Default is TD_OFF. You must enable the driver tracing level before you can enable the infrastructure tracing level.
Trace File Name	File name and path of the Teradata PT API trace file. Default path is $PM_HOME. Default file name is <Name of the TPT Operator>_timestamp. For example, EXPORTER_20091221.
Query Band Expression	The query band expression to be passed to the Teradata PT API. A query band expression is a set of name-value pairs that identify a query's originating source. In the expression, each name-value pair is separated by a semicolon and the expression ends with a semicolon. For example, ApplicationName=Informatica;Version=9.0.1;ClientUser=A;.
Spool Mode	Determines the spool mode Teradata PT API uses to extract data from Teradata. You can select one of the following spool modes: - Spool. Teradata PT API spools data while extracting data from Teradata. - NoSpool. Teradata PT API does not spool data while extracting data from Teradata. If the database does not support the NoSpool option, Teradata PT API uses the Spool option. - NoSpoolOnly. Teradata PT API does not spool while extracting data from Teradata. Default is Spool.
SQL Override Query	The SQL statement to override the default query used to read data from the Teradata source. The data type, number, and order of columns in the select clause must match with the Teradata source object.
Select Distinct	Selects distinct rows and eliminates duplicate rows.
Tracing Level	Determines the amount of detail that appears in the log for the source. you can choose Terse, Normal, Verbose Initialization, or Verbose Data tracing level. Default is Normal.

Partitioning for Teradata Sources

When you read data from Teradata sources, you can configure partitioning to read data in parallel and optimize the mapping performance at run time.

The Secure Agent distributes rows of data based on the number of partitions you define. Click the **Partitions** tab in the Source transformation to define the number of partitions you want to use to read data in parallel.

Rules and Guidelines for Partitioning

Consider the following rules and guidelines when you configure partitioning for Teradata sources:

- If the mapping contains multiple Teradata sources, configure the same number of partitions for all the Teradata sources. Otherwise, the mapping validation fails.
- If you configure more than one partition for a Teradata source, you cannot write data to a Teradata target.

Teradata Targets in Mappings

In a mapping, you can configure a Target transformation to represent a single Teradata API target.

The following table describes the Teradata target properties that you can configure in a Target transformation:

Property	Description
Connection	Select the target connection, or click **New Connection** to create one. Alternatively, you can define a connection parameter in the mapping and enter a specific connection in each mapping task that is associated with the mapping.
Target Type	Type of the target object. Select Single Object or Parameter.
Object	Name of the target object. Select one of the following types: - Existing. Select to specify an existing target. - Create New at Runtime. Select to create the target object when you run the task. Enter the name of the object, and then click **OK**. If you select Parameter, create or select the parameter for the target object.
Operation	Target operation. Select Insert, Update, Upsert, Delete, or Data Driven. **Note:** When you define a DD_UPDATE expression in the data driven condition, ensure that you map the NOT NULL columns.

The following table describes the Teradata target advanced properties that you can configure in a Target transformation:

Property	Description
Update Else Insert	Teradata PT API updates existing rows and inserts other rows as if marked for update. If disabled, Teradata PT API updates existing rows only. The Secure Agent ignores this attribute when you treat source rows as inserts or deletes. Default is disabled.
Truncate Table	Teradata PT API deletes all rows in the Teradata target before it loads data. Default is disabled.

Property	Description
Mark Missing Rows	Specifies how Teradata PT API handles rows that do not exist in the target table: - None. If Teradata PT API receives a row marked for update or delete but it is missing in the target table, Teradata PT API does not mark the row in the error table. - For Update. If Teradata PT API receives a row marked for update but it is missing in the target table, Teradata PT API marks the row as an error row. - For Delete. If Teradata PT API receives a row marked for delete but it is missing in the target table, Teradata PT API marks the row as an error row. - Both. If Teradata PT API receives a row marked for update or delete but it is missing in the target table, Teradata PT API marks the row as an error row. Default is None.
Mark Duplicate Rows	Specifies how Teradata PT API handles duplicate rows when it attempts to insert or update rows in the target table: - None. If Teradata PT API receives a row marked for insert or update that causes a duplicate row in the target table, Teradata PT API does not mark the row in the error table. - For Insert. If Teradata PT API receives a row marked for insert but it exists in the target table, Teradata PT API marks the row as an error row. - For Update. If Teradata PT API receives a row marked for update that causes a duplicate row in the target table, Teradata PT API marks the row as an error row. - Both. If Teradata PT API receives a row marked for insert or update that causes a duplicate row in the target table, Teradata PT API marks the row as an error row. This attribute is available for the Update and Stream system operators. Default is For Insert.
Mark Extra Rows	Specifies how Teradata PT API marks error rows when it attempts to update or delete multiple rows in the target table: - None. If Teradata PT API receives a row marked for update or delete that affects multiple rows in the target table, Teradata PT API does not mark the row in the error table. - For Update. If Teradata PT API receives a row marked for update that affects multiple rows in the target table, Teradata PT API marks the row in the error table. - For Delete. If Teradata PT API receives a row marked for delete that affects multiple rows in the target table, Teradata PT API marks the row in the error table. - Both. If Teradata PT API receives a row marked for update or delete that affects multiple rows in the target table, Teradata PT API marks the row in the error table. Default is Both.
Log Database	Name of the database that stores the log tables.
Log Table Name	Name of the restart log table.
Error Database	Name of the database that stores the error tables.
Error Table Name1	Name of the first error table.
Error Table Name2	Name of the second error table.
Work Table Database	Name of the database that stores the work tables.

Property	Description
Work Table Name	Name of the work table.
Macro Database	Name of the database that stores the macros Teradata PT API creates when you select the Stream system operator. The Stream system operator uses macros to modify tables. It creates macros before Teradata PT API begins loading data and removes them from the database after Teradata PT API loads all rows to the target. If you do not specify a macro database, Teradata PT API stores the macros in the log database.
Drop Log/Error/ Work Tables	Drops existing log, error, and work tables for a session when the session starts. Default is disabled.
Serialize	Uses the Teradata PT API serialize mechanism to reduce locking overhead when you select the Stream system operator. Default is enabled.
Pack	Number of statements to pack into a request when you select the Stream system operator. Must be a positive, nonzero integer. Default is 20. Minimum is 1. Maximum is 600.
Pack Maximum	Causes Teradata PT API to determine the maximum number of statements to pack into a request when you select the Stream system operator. Default is disabled.
Buffers	Determines the maximum number of request buffers that may be allocated for the Teradata PT API job when you select the Stream system operator. Teradata PT API determines the maximum number of request buffers according to the following formula: `Max_Request_Buffers = Buffers * Number_Connected_Sessions` Must be a positive, nonzero integer. Default is 3. Minimum is 2.
Error Limit	Maximum number of records that can be stored in the error table before Teradata PT API terminates the Stream system operator job. Must be -1 or a positive, nonzero integer. Default is -1, which specifies an unlimited number of records.
Replication Override	Specifies how Teradata PT API overrides the normal replication services controls for an active Teradata PT API session: - On. Teradata PT API overrides normal replication services controls for the active session. - Off. Teradata PT API disables override of normal replication services for the active session when change data capture is active. - None. Teradata PT API does not send an override request to the Teradata Database. Default is None.

Property	Description
Driver Tracing Level	Determines Teradata PT API tracing at the driver level: - TD_OFF. Teradata PT API disables tracing. - TD_OPER. Teradata PT API enables tracing for driver-specific activities for Teradata. - TD_OPER_ALL. Teradata PT API enables all driver-level tracing. - TD_OPER_CLI. Teradata PT API enables tracing for activities involving CLIv2. - TD_OPER_NOTIFY. Teradata PT API enables tracing for activities involving the Notify feature. - TD_OPER_OPCOMMON. Teradata PT API enables tracing for activities involving the operator common library. Default is TD_OFF.
Infrastructure Tracing Level	Determines Teradata PT API tracing at the infrastructure level: - TD_OFF. Teradata PT API disables tracing. - TD_OPER. Teradata PT API enables tracing for driver-specific activities for Teradata. - TD_OPER_ALL. Teradata PT API enables all driver-level tracing. - TD_OPER_CLI. Teradata PT API enables tracing for activities involving CLIv2. - TD_OPER_NOTIFY. Teradata PT API enables tracing for activities involving the Notify feature. - TD_OPER_OPCOMMON. Teradata PT API enables tracing for activities involving the operator common library. Default is TD_OFF. You must enable the driver tracing level before you can enable the infrastructure tracing level.
Trace File Name	File name and path of the Teradata PT API trace file. Default path is $PM_HOME. Default file name is <Name of the TPT Operator>_timestamp. For example, LOAD_20091221.
Pause Acquisition	Causes load operation to pause before the session loads data to the Teradata PT API target. Disable when you want to load the data to the target. Default is disabled.
Query Band Expression	The query band expression to be passed to the Teradata PT API. A query band expression is a set of name-value pairs that identify a query's originating source. In the expression, each name-value pair is separated by a semicolon and the expression ends with a semicolon. For example, ApplicationName=Informatica;Version=9.0.1;ClientUser=A;.
Serialize Columns	Specifies an ordered list of columns that need to be serialized for the stream operator. Separate each column by semicolon. Use this option to serialize based on a single column or set of columns. You can specify a value when you enable the serialize mechanism. Default is blank. You can specify a value when you enable the serialize mechanism.
Replacement Character	Character to use in place of an unsupported Teradata unicode character in the Teradata database while loading data to targets. You can enter one character.
Database Version	Teradata database version. If you specified a character used in place of an unsupported character while loading data to Teradata targets, specify the version of the target Teradata database. Use this attribute in conjunction with the Replacement Character attribute. The Secure Agent ignores this attribute if you did not specify a replacement character while loading data to Teradata targets. Default is 8x-13x.

Property	Description
System Operator	The Teradata PT API operator type. You can select one of the following options: - Load. Bulk loads data into an empty Teradata database table. - Update. Performs update, insert, upsert, and delete operations against Teradata database tables. - Stream. Performs update, insert, upsert, and delete operations against Teradata database tables in near real-time mode. Default is Stream.
Write Buffer Size	This property is applicable when you use the Load operator. Defines the maximum buffer size in kilobytes to be allocated to the Teradata PT API job for writing data. If you do not enter a value and you use TTU version 15.10 or 16.10, Teradata Connector sets the value as 1024 KB for optimal performance. This setting is in line with Teradata's recommendation.
Success File Directory	Not applicable.
Error File Directory	Not applicable.
Forward Rejected Rows	Not applicable.

Partitioning for Teradata Targets

When you use a Teradata connection to run a mapping to write data to Teradata, you can configure partitioning for the Teradata target.

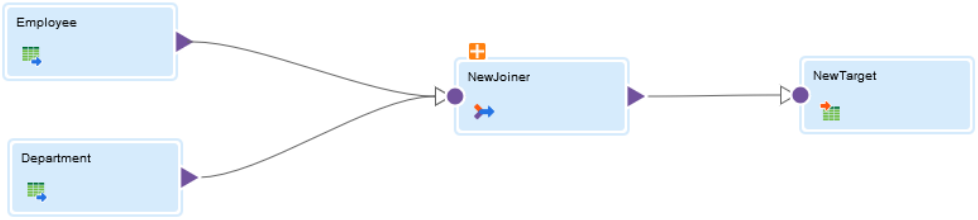
You do not have to specify the number of partitions for the target separately. The Secure Agent considers the same number of partitions for the target as the number defined for the source partitioning.

Mapping Example

You want to read the employee and department information from a Teradata source, join the data, and then load the data to a Teradata target table.

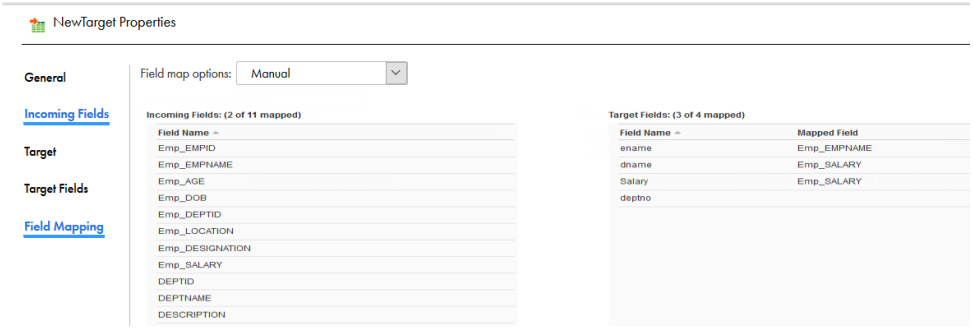
Perform the following tasks to configure a mapping task:

1. Create a Teradata mapping and enter a name and description.
   The following image shows the Teradata mapping:



2. Add a Source transformation to read employee information from Teradata using a Teradata connection.
   The source object used is a single object named Employee.
   Some of the employee fields included are EMPID, EMPNAME, AGE, and DOB.

3. Add another Source transformation to read the department information using a Teradata connection. The source object is a single object named Department.
   Some of the business fields included are DEPTID, DEPTNAME, and DESCRIPTION.
4. Add a Joiner transformation to join the employee and department information. Use the following Join condition for Master, Operator, and Detail:
   `Emp_DEPTID = DEPTID`
5. Add a Target transformation to write the joined data to a Teradata table using the Teradata connection. Perform the following tasks:
   a. Select the Teradata connection.
   b. Use the insert operation and use a single target object named emp_dept_tgt.
   c. In the Advanced Properties section, select Stream system operator and specify the data base version to which you want to connect.
   d. In the Field Mapping tab, select the incoming fields that you want in the target.
   e. The following image shows the mapped fields:



6. Configure a Teradata mapping task, add the mapping, and run the task.
   The Secure Agent updates the target Teradata table.

Calling a stored procedure

When you use the ODBC connection and select the **ODBC subtype** as **Teradata**, you can use an SQL transformation to call a stored procedure in Teradata. You can use the SQL transformation to process SQL queries midstream in a pipeline.

You can configure the SQL transformation to process the following types of SQL statements:

Stored procedure

Stored procedures reside in the database and run within the database. When you configure the SQL transformation to process a stored procedure, it passes input parameters to the stored procedure. The stored procedure passes the return value or values to the output fields of the transformation.

Note: Stored procedure definitions must not contain keywords, special characters, and Unicode characters.

SQL Query

You can configure the SQL transformation to process a saved query that you create in Data Integration or you can enter a query in the SQL editor.

Note: Stored function is not supported.

For more information about SQL transformations, see *Transformations* in the Data Integration help.

CHAPTER 5

Teradata ODBC SQL ELT optimization

You can configure Full and Source SQL ELT optimization for the ODBC connection type that uses Teradata ODBC drivers for mapping.

When you configure SQL ELT optimization, the Secure Agent pushes the transformation logic to the source or target databases. Use SQL ELT optimization when you use database resources to improve the performance of the task.

When you run a task configured for SQL ELT optimization, the task converts the transformation logic to an SQL query. The task sends the query to the database, and the database executes the query.

You cannot use SQL ELT optimization when you run a mapping in advanced mode.

Supported transformations

When you configure SQL ELT optimization, the Secure Agent tries to push the configured transformation to the database.

The following table lists the supported SQL ELT types for the Teradata database:

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

Supported functions

When you enable 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 functions in Teradata:

Functions	Supported SQL ELT type
ABS()	Source, Full
ADD_TO_DATE()	Full
AVG()	Source, Full
CEIL()	Full
CONCAT()	Full
COS()	Source, Full
COSH()	Full
COUNT()	Source, Full
DATE_COMPARE()	Source, Full
DECODE()	Source, Full
EXP()	Source, Full
FLOOR()	Full
GET_DATE_PART()	Full
IIF()	Source, Full
IN()	Source, Full
INSTR()	Full
ISNULL()	Source, Full
LENGTH()	Full
LN()	Full
LOG()	Full
LOWER()	Source, Full
LTRIM()	Full
MAX()	Source, Full

Functions	Supported SQL ELT type
MIN()	Source, Full
MOD()	Full
POWER()	Source, Full
ROUND(NUMBER)	Full
RTRIM()	Full
SIGN()	Full
SIN()	Source, Full
SQRT()	Source, Full
STDDEV()	Full
SUBSTR()	Full
SUM()	Source, Full
SYSTIMESTAMP()	Full
TAN()	Source, Full
TANH()	Full
TO_BIGINT	Full
TO_CHAR(DATE)	Full
TO_CHAR(NUMBER)	Full
TO_DATE()	Full
TO_DECIMAL()	Full
TO_FLOAT()	Full
TO_INTEGER()	Full
TRUNC(NUMBER)	Full
UPPER()	Source, Full
VARIANCE()	Full

Supported 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 variables in Teradata:

Variables	Supported SQL ELT Type
SESSSTARTTIME	Full
SYSDATE	Full

Configuring the Teradata ODBC driver

Teradata supports Teradata ODBC drivers on Windows and Linux systems. You must install the Teradata ODBC 64-bit driver based on your system requirement.

Configuring Teradata ODBC driver on Windows

Before you establish an ODBC connection to connect to Teradata on Windows, you must configure the Teradata ODBC drivers.

Perform the following tasks to configure the Teradata ODBC driver on Windows:

1. Download and install the Teradata ODBC 64-bit drivers on the machine where the Secure Agent is installed.
2. Run the `odbcad32.exe` file.
3. In the ODBC Data Source Administrator dialog box, click **System DSN**.
4. Configure the applicable Teradata connection settings.

Configuring the Teradata ODBC driver on Linux

Before you can run tasks to connect to Teradata using the ODBC connection from Linux, you must set the ODBCINI and LD_LIBRARY_PATH environment variables for the driver and create the DSN entries.

1. Add the path of the `odbc.ini` file to the ODBCINI environment variable. For example,

   ```
   setenv ODBCINI "/data/home/adputf_9/cloud_td/ODBCINI/odbc.ini"
   ```

2. To set the LD_LIBRARY_PATH environment variable, use the following syntax:

   ```
   setenv LD_LIBRARY_PATH "/opt/teradata/client/<Version>/lib64"
   ```

3. Add entries for the Teradata data sources in the `odbc.ini` file.

   The following section shows a sample entry in the `odbc.ini` file:

   ```
   [Sample Teradata ODBC DSN]
   [ODBC Data Sources]
   <DSN_NAME>=tdata.so
   ```

```
[<DSN_NAME>]
Driver=<Teradata_ClientHome>/lib64/tdata.so
Description=DataDirect 7.1 Teradata
AccountString=
AuthenticationDomain=
AuthenticationPassword=
AuthenticationUserid=
CharacterSet=ASCII
DBCName=<Teradata Server>
Database=
EnableDataEncryption=0
EnableExtendedStmtInfo=0
EnableLOBs=1
EnableReconnect=0
IntegratedSecurity=0
LoginTimeout=20
LogonID=
MapCallEscapeToExec=0
MaxRespSize=8192
Password=
PortNumber=1025
PrintOption=N
ProcedureWithSplSource=Y
ReportCodePageConversionErrors=0
SecurityMechanism=
SecurityParameter=
ShowSelectableTables=1
TDProfile=
TDRole=
TDUserName=
```

4. Restart the Secure Agent after you configure the environment variables.

Configuring a Teradata ODBC connection

You must create an ODBC connection to connect to Teradata after you configure the Teradata ODBC drivers. You can then configure SQL ELT optimization for the mapping that uses the ODBC connection type to enhance the mapping performance.

When you create a Teradata ODBC connection, specify the required Teradata database details in the ODBC connection properties.

You must select the **ODBC subtype** as **Teradata** in the ODBC connection properties.

If you connect to Teradata from Linux, you must select **Data Direct** as the **Driver Manager for Linux** in the ODBC connection properties.

Configuring SQL ELT optimization

Perform the following steps to configure SQL ELT optimization for Teradata sources or targets:

1. Create a mapping task.
2. In the **SQL ELT Optimization** section on the **Runtime Options** tab, select the SQL ELT type.
3. From the **SQL ELT Optimization** list, select the required type of SQL ELT optimization.

When you run the mapping task, the transformation logic is pushed to the configured database. To verify that the mapping was optimized, you can check the session log for the job. In Monitor, view the log for jobs.

Configuring cross-schema SQL ELT optimization for a Teradata ODBC mapping

You can use cross-schema SQL ELT optimization for a mapping to read or write data based on different schemas within the same database.

To configure cross-schema SQL ELT optimization for a Teradata ODBC mapping task, perform the following steps:

1. Create Teradata ODBC source and target connections, each defined with a different schema.

   For example,

   - Create a `teradata_odbc1` Teradata ODBC connection and specify the `TERADATA_SCHEMA1` schema in the connection properties.
   - Create a `teradata_odbc2` Teradata ODBC connection and specify the `TERADATA_SCHEMA2` schema in the connection properties.
2. Create a Teradata ODBC mapping.

   For example, create a `m_teradata_pdo_crossSchema` Teradata ODBC mapping.
3. Add a Source transformation and include a Teradata object and connection to read data using the schema specified in the connection.

   For example, add a Source transformation and include a Teradata source object and connection `teradata_odbc1` to read data using `TERADATA_SCHEMA1`.
4. Add a Target transformation and include a Teradata target object and connection to write data using the schema specified in the connection.

   For example, add a Target transformation and include a Teradata target object and connection `teradata_odbc2` to write data using `TERADATA_SCHEMA2`.
5. Create a Teradata ODBC mapping task, and perform the following tasks:

   a. Select the configured Teradata ODBC mapping.

      For example, select the `m_teradata_pdo_crossSchema` Teradata ODBC mapping.

   b. In the **SQL ELT Optimization** section on the **Runtime Options** tab, set the SQL ELT optimization value to **Full**.

   c. In the **Advanced Session Properties** section, select the **Enable cross-schema SQL ELT optimization** check box.

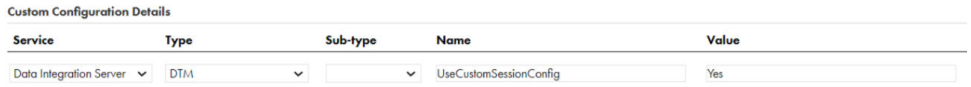
   d. Save the task, and click **Finish**.

   When you run the mapping task, the Secure Agent reads data from the Teradata source object associated with the `TERADATA_SCHEMA1` schema and writes data to the Teradata target object associated with the `TERADATA_SCHEMA2` schema.

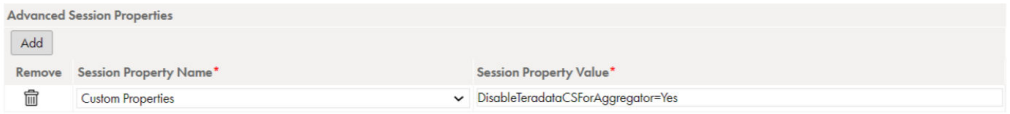
Enabling SQL ELT optimization for Expression and Aggregator transformations

Before you enable SQL ELT optimization for a Teradata mapping task that includes Expression and Aggregator transformations in the mapping, you need to set certain properties for the Secure Agent and the task.

1. In **Administrator**, perform the following tasks in the Secure Agent properties:
   a. Select the Secure Agent listed on the **Runtime Environments** tab, and click **Edit**.
   b. In the **Custom Configuration Details** section, select **Data Integration Service** as the service and **DTM** as the type.
   c. Set *UseCustomSessionConfig* to *Yes*.



2. In Data Integration, perform the following steps in the mapping task:
   a. In the **Runtime Options** page, navigate to **Advanced Session Properties**.
   b. Select **Custom Properties** in the session property name list and enter the following value: *DisableTeradataCSForAggregator=Yes*



Rules and guidelines for SQL ELT optimization

Consider the following rules and guidelines when you configure full SQL ELT optimization to a Teradata database using the Teradata ODBC connection:

- You cannot push the LTRIM(), RTRIM(), or ROUND(NUMBER) function that contains more than one argument to the Teradata database.
- You cannot use the upsert operation in a Joiner transformation.
- When you configure an SQL override query using full SQL ELT optimization, you must map all fields that you specify in the SQL override query to the Teradata target object.
- You can push the STDDEV() and VARIANCE() functions to the Teradata database only in an Aggregator transformation.
- You cannot use a ORDER BY clause in a custom query or SQL override query, unless you also specify the TOP clause in the query.

CHAPTER 6

Data Type Reference

Data Integration uses the following data types in mappings and mapping task with Teradata.

Teradata Native Data types

Teradata datatypes 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 run-time environment uses to move data across platforms. Transformation data types appear in all transformations in mappings and mapping tasks.

Teradata Data Types and Transformation Data Types

The following table lists the Teradata data types that the run-time environment supports and the corresponding transformation data types:

Teradata Data Type	Range	Transformation Data Type	Range
Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
Byte	1 to 64,000 bytes	Binary	1 to 104,857,600 bytes
Byteint	-128 to 127	Integer	Precision 10, scale 0
Char	1 to 64,000 bytes	String	1 to 104,857,600 characters
Date	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. Precision 19, scale 0	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
Decimal	Precision 1 to 18, scale 0 to 18	Decimal	Precision 1 to 28, scale 0 to 28
Float	-2.226E+308 to 1.797E+308	Double	Precision 15

Teradata Data Type	Range	Transformation Data Type	Range
Integer	-2,147,483,648 to 2,147,483,647	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Smallint	-32768 to 32768	Integer	Precision 10, scale 0
Time	00:00:00.000000 to 23:59:61.999999 Precision 15, scale 6	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
Timestamp	1 to 19 characters Precision 19 to 26, scale 0 to 6	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
Varbyte	1 to 64,000 bytes	Binary	1 to 104,857,600 bytes
Varchar	1 to 64,000 bytes	String	1 to 104,857,600 characters

INDEX

C

Cloud Application Integration community
  URL 5
Cloud Developer community
  URL 5
configuring Teradata ODBC driver
  Linux 31
connections
  Teradata 8, 10

D

Data Integration community
  URL 5

E

error tables
  Teradata PT, description 18

F

functions
  SQL ELT optimization 29

I

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

L

log tables
  Teradata PT API, description 18

M

maintenance outages 6
mappings
  Teradata 20
  Teradata source properties 20
  Teradata target properties 22

N

native data types 35

R

row-level processing
  duplicate rows 17
  extra rows 17
  missing rows 17

S

Snowflake ODBC connection 32
Snowflake ODBC driver 32
Source transformation
  Teradata properties 20
sources
  Teradata in mappings 20
SQL ELT optimization
  activity log 32
  advanced session properties 32
  configuration 32
  full pushdown 28
  functions 29
  source pushdown 28
  target pushdown 28
  transformations 28
  variables 31
status
  Informatica Intelligent Cloud Services 6
system status 6

T

Target transformation
  Teradata properties 22
targets
  Teradata in mappings 22
Teradata
  connection properties 10
  mappings 20
  Source transformation 20
  sources in mappings 20
  Target transformation 22
  targets in mappings 22
Teradata connector
  rules and guidelines 34
Teradata Connector
  assets 7
  overview 8
Teradata data types
  mapping with transformation data types 35
Teradata PT API system operators
  load 16
  stream 16
  update 16
transformation data types 35

transformations
SQL ELT optimization 28
trust site
description 6

U

upgrade notifications 6

V

variable
SQL ELT optimization 31

W

web site 5
work tables
Teradata PT API, description 19