



Informatica® PowerExchange for Teradata
Parallel Transporter API

10.5.9

User Guide for PowerCenter

© Copyright Informatica LLC 2010, 2025

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.

Informatica, the Informatica logo, PowerCenter, and PowerExchange 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.

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.

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

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: 2025-09-03

Table of Contents

Preface	4
Informatica Resources.	4
Informatica Network.	4
Informatica Knowledge Base.	4
Informatica Documentation.	4
Informatica Product Availability Matrices.	5
Informatica Velocity.	5
Informatica Marketplace.	5
Informatica Global Customer Support.	5
 Chapter 1: Understanding PowerExchange for Teradata Parallel Transporter API.....	 6
Understanding PowerExchange for Teradata Parallel Transporter API Overview.	6
 Chapter 2: Configuration.....	 8
Configuration Overview.	8
Prerequisites.	8
Setting Environment Variables.	9
Verify ODBC Connectivity.	10
 Chapter 3: Teradata PT API Sessions and Workflows.....	 11
Configuring a Teradata Parallel Transporter API Connection.	11
Creating a Teradata PT Connection Object.	11
Working with Teradata PT API Sessions.	14
Configuring Recovery.	14
Working with Log, Error, and Work Tables.	15
Pipeline Partitioning.	16
Support for Multiple Instances.	17
Staged Loading.	17
Spool Modes.	18
Configuring a Session for a Teradata PT API Source.	18
Configuring a Session for a Teradata PT API Target.	19
 Appendix A: Datatype Reference.....	 25
Datatype Reference Overview.	25
Teradata and Transformation Datatypes.	25
Datatype Synonyms.	26
 Index.....	 27

Preface

Use the *Informatica® PowerExchange® for Teradata Parallel Transporter API User Guide* to learn how to read from and write to Teradata by using the PowerCenter Client. Learn to create a Teradata Parallel Transporter API connection, develop mappings, and run sessions in an Informatica domain. This guide also includes information on pushdown optimization, partitioning, and parameterization of Teradata sources and targets.

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 Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

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 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 Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at <http://velocity.informatica.com>. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <https://network.informatica.com> and select the eSupport option.

CHAPTER 1

Understanding PowerExchange for Teradata Parallel Transporter API

This chapter includes the following topic:

- [Understanding PowerExchange for Teradata Parallel Transporter API Overview, 6](#)

Understanding PowerExchange for Teradata Parallel Transporter API Overview

PowerExchange for Teradata Parallel Transporter API integrates PowerCenter and Teradata Parallel Transporter API (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. You can create a PowerCenter session that uses Teradata PT API to extract from Teradata sources and load to Teradata targets.

To use PowerExchange for Teradata Parallel Transporter API, create a mapping with a Teradata source or target. To use PowerExchange for Teradata Parallel Transporter API, create a mapping with a Teradata source or target. Use a Teradata PT connection to connect to the Teradata tables you want to export or load in a session.

The PowerCenter Integration Service uses the Teradata PT API infrastructure to connect to Teradata. It extracts or loads data using one of the following system operators:

- **Export.** Extracts data from Teradata.
- **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.

You cannot access the following Teradata objects:

- Macro
- Trigger
- Joined index
- Stored procedures

- Hash index
- Permanent journal

CHAPTER 2

Configuration

This chapter includes the following topics:

- [Configuration Overview, 8](#)
- [Prerequisites, 8](#)
- [Setting Environment Variables, 9](#)
- [Verify ODBC Connectivity, 10](#)

Configuration Overview

PowerExchange for Teradata Parallel Transporter API installs with PowerCenter.

To configure PowerExchange for Teradata Parallel Transporter API, perform the following steps:

1. Complete the prerequisites.
2. Set system environment variables.
3. Verify ODBC connectivity.

Prerequisites

Complete the following tasks if you want to use PowerExchange for Teradata Parallel Transporter API:

- Install or upgrade PowerCenter. Use the server installer to install the Informatica services. Use the client installer to install the PowerCenter Client. PowerExchange for Teradata Parallel Transporter API is installed when you install PowerCenter.
- Install the Teradata Parallel Transporter utilities on the machines where the PowerCenter Integration Service runs.

The following table lists the Teradata Parallel Transporter utilities:

Teradata Parallel Transporter Utilities
Teradata Parallel Transporter Base
Teradata Parallel Transporter Stream Operator
Teradata CLv2
Teradata ODBC
Teradata Generic Security Services ¹
Shared ICU Libraries for Teradata

¹. For UNIX platforms, verify that you have installed Teradata Generic Security Services.

Setting Environment Variables

You must configure Java and Teradata environment variables before you can use PowerExchange for Teradata Parallel Transporter API.

The following table describes the environment variables that you can set:

Environment Variable	Value
JAVA_HOME	<Informatica installation directory>/java
JRE_HOME	\${JAVA_HOME}/jre
PATH	\${JAVA_HOME}/bin:\${JRE_HOME}/bin:\${PATH}

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

Environment Variable	Value
THREADONOFF	On UNIX and Linux operating systems, set the <code>THREADONOFF</code> environment variable to 1 to enable multithreading support for Teradata Parallel Transporter processes.
NLSPATH	Set to the location of the <code>opermsgs.cat</code> file. For example, <code>/opt/teradata/client/<version>/msg/%N:/opt/teradata/client/<version>/odbc_64/msg/%N:/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/%L/%N.cat</code>

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
AIX	LIBPATH

For example, use the following syntax for Linux:

- Using a Bourne shell:

```
$ LD_LIBRARY_PATH=${JAVA_HOME}/lib:/opt/teradata/client/<API version>/lib:/opt/teradata/client/<API version>/lib64:/usr/lib64:/usr/lib:${ORACLE_HOME}/lib:${DB2DIR}/lib64; export LD_LIBRARY_PATH
```

- Using a C shell:

```
$ setenv LD_LIBRARY_PATH ${JAVA_HOME}/lib:/opt/teradata/client/<API version>/lib:/opt/teradata/client/<API version>/lib64:/usr/lib64:/usr/lib:${ORACLE_HOME}/lib:${DB2DIR}/lib64; export LD_LIBRARY_PATH
```

For example, use the following syntax for AIX:

- Using a Bourne shell:

```
$ LIBPATH=${LIBPATH}:${JAVA_HOME}/lib:/opt/teradata/client/<API version>/lib:/opt/teradata/client/<API version>/lib64:/usr/lib64:/usr/lib:${ORACLE_HOME}/lib:${DB2DIR}/lib64; export LIBPATH
```

- Using a C shell:

```
$ setenv LIBPATH ${JAVA_HOME}/lib:/opt/teradata/client/<API version>/lib:/opt/teradata/client/<API version>/lib64:/usr/lib64:/usr/lib:${ORACLE_HOME}/lib:${DB2DIR}/lib64; ${LIBPATH}
```

Verify ODBC Connectivity

Verify the ODBC connectivity on the host where you installed the Informatica services.

Use the following command to test the ODBC connection:

```
ssgodbc.linux64 -d "DSN" -u "username" -p "password" -v
```

The ssgodbc utility is available in the following directory:

```
<Informatica installation directory>/tools/debugtools/ssgodbc
```

CHAPTER 3

Teradata PT API Sessions and Workflows

This chapter includes the following topics:

- [Configuring a Teradata Parallel Transporter API Connection, 11](#)
- [Working with Teradata PT API Sessions, 14](#)
- [Configuring a Session for a Teradata PT API Source, 18](#)
- [Configuring a Session for a Teradata PT API Target, 19](#)

Configuring a Teradata Parallel Transporter API Connection

Create Teradata PT API connection objects to define connections to Teradata PT API in the repository. The PowerCenter Integration Service uses Teradata PT API connection objects to extract from or load to Teradata.

Additionally, you can create a Teradata target ODBC connection. The PowerCenter Integration Service uses the target ODBC connection to drop log, error, and work tables, truncate target tables, and create recovery tables in the target database. The PowerCenter Integration Service does not use the ODBC connection to extract from or load to Teradata.

If you enable recovery in a session that loads to Teradata, create a Teradata target ODBC connection or create the recovery table manually.

Creating a Teradata PT Connection Object

You configure Teradata Parallel Transporter API properties when you create a Teradata PT API connection. You can override the system operator in the session properties.

For more information about Teradata PT API settings, see the Teradata Parallel Transporter documentation.

To create a Teradata Parallel Transporter connection:

1. Click **Connections > Relational** in the Workflow Manager.
The **Relational Connection Browser** dialog box appears.
2. Click **New**.
The **Select Subtype** dialog box appears.

3. Select **Teradata PT Connection**, and then click **OK**.
The **Relational Connection Editor** dialog box appears.
4. Specify the properties for the connection and the connection attributes.
5. Click **OK**.

PowerExchange for Teradata Parallel Transporter Connections

When you configure a Teradata Parallel Transporter API connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to Teradata.

The following table describes the Teradata Parallel Transporter API connection properties that you must configure:

Property	Description
Name	Connection name used by the Workflow Manager. Connection name cannot contain spaces or other special characters, except for the underscore.
User Name	Database user name with the appropriate read and write database permissions to access the database. To define the user name in the parameter file, enter session parameter <i>\$ParamName</i> as the user name, and define the value in the session or workflow parameter file. The PowerCenter Integration Service interprets user names that start with <i>\$Param</i> as session parameters. You can connect to a database that runs on a network that uses Kerberos authentication. To configure Kerberos authentication for the database connection, set the user name to the reserved word <i>PmKerberosUser</i> . If you use Kerberos authentication, the connection uses the credentials of the user account that runs the session to connect to the database. The user account must have a user principal on the Kerberos network where the database runs.
Use Parameter in Password	Indicates the password for the database user name is a session parameter, <i>\$ParamName</i> . If you enable this option, define the password in the workflow or session parameter file, and encrypt it using the <i>pmpasswd</i> CRYPT_DATA option. Default is disabled.
Password	Password for the database user name. If you set the user name to <i>PmKerberosUser</i> to use Kerberos authentication for the database connection, set the password to the reserved word <i>PmKerberosPassword</i> . The connection uses the credentials of the user account that runs the session to connect to the database.
Code Page	Code page associated with the database. When you run a session that extracts 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.

The following table describes the Teradata connection properties that you must configure:

Attribute	Description
TDPID	The name of the Teradata database machine.
Database Name	Teradata database name. If you do not enter a database name, Teradata PT API uses the default login database name.

Attribute	Description
Tenacity	Amount of time, in hours, that Teradata PT API continues trying to log on when the maximum number of operations runs on the Teradata database. Must be a positive integer. Default is 4.
Max Sessions	Maximum number of sessions that Teradata PT API establishes with the Teradata database. Must be 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. Must be a positive integer between 1 and the Max Sessions value. Default is 1.
Sleep	Amount of time, in minutes, that Teradata PT API pauses before it retries to log on when the maximum number of operations runs on the Teradata database. Must be 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 or write the data block size from source or target through the Export operator. Maximum is 16775168 bytes for Teradata Database version 16.20 and higher. If the Teradata Database version is lower than 16.20, then 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 Teradata logs and verify the Teradata documentation of the same version.
Authentication Type	Method to authenticate the user. Select one of the following authentication types: <ul style="list-style-type: none"> - Native. Authenticates your user name and password against the Teradata database specified in the connection. - LDAP. Authenticates user credentials against the external LDAP directory service. - KRB5. Authenticates the credentials of the user account that runs the session against the Kerberos network where the database runs. Default is Native.
System Operator	Teradata PT API operator type: <ul style="list-style-type: none"> - Export. Extracts data from Teradata. Select Export if the session uses a Teradata Parallel Transporter Reader. - 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. Select Stream if you want to enable recovery for sessions that load data to Teradata. Default is Stream.

Working with Teradata PT API Sessions

When you configure a Teradata PT API session, you define properties that determine how the PowerCenter Integration Service extracts data from or loads data to Teradata sources and targets. You can configure the following Teradata PT API session properties:

- Recovery
- Error and log tables
- Pipeline partitioning

Configuring Recovery

You can recover Teradata PT API sessions that meet the following requirements:

- The session must be enabled for recovery.
- The session must contain a PowerExchange for CDC source.
- The session must be configured to use the Stream system operator to load data to Teradata.
- Set the `TPTRecoveryModel` custom property to “yes” for the session or PowerCenter Integration Service.
- If the PowerCenter Integration Service runs on UNIX, set the `THREADONOFF` environment variable to 1.

A recovery table must exist in the target database before the PowerCenter Integration Service can recover a session or workflow. You can either create the recovery table manually or configure a target ODBC connection in addition to the Teradata PT API target connection object. Configure a target ODBC connection when you want to drop log, error, and work tables or truncate target tables. Otherwise, you can create the recovery table manually.

Creating the Recovery Table using a Target ODBC Connection

When you configure a Teradata PT API target connection and a target ODBC connection in a session, the PowerCenter Integration Service creates the recovery table in the target database.

You can also configure the following session properties:

- **Drop log, error, and work tables.** Log tables can contain inconsistent data when a session fails. If you configure the session to use a Teradata PT API target connection and a target ODBC connection, the PowerCenter Integration Service can drop existing log, error, and work tables for a session when the session starts.
- **Truncate the target tables.** Teradata PT API can truncate the target tables when the system operator is set to Update or Stream in the connection object properties. If you configure the session to use a Teradata PT API target connection and a target ODBC connection, Teradata PT API can truncate the target tables when the system operator is Update, Stream, or Load.

Creating the Recovery Table Manually

If you enable recovery for a session that loads to Teradata, and you do not need to drop log, error, and work tables or truncate target tables, you can create the recovery table manually. Create the recovery table in the Teradata target database.

To create the recovery table, execute the `create_schema_tpt.sql` script against the Teradata target database. The script is installed in the following directory:

```
<PowerCenter Installation Directory>\server\bin\RecoverySQL
```

The script contains the following query:

```
CREATE SET TABLE INFARecoveryTable
(
    TPTCPData      INTEGER,
    Counter         INTEGER,
    RecInfo         VARCHAR(256),
    CheckpointNum   INTEGER,
    FrameworkInfo   VARBYTE(2000)
    PRIMARY INDEX   (RecInfo);
```

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.** Store Teradata PT API restart and log information. Teradata PT API creates one log table for each partition.
- **Error Tables.** Log Teradata errors and rejected data when a session runs. Teradata PT API creates two error tables for each partition.
- **Work Tables.** Store 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 session 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 session to load to Teradata.

The following table describes the session properties that allow you to specify 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 session 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 PowerCenter Integration Service uses the name <code><log_database>.INFA_LT_<number></code> . The exact table name appears in the session log.

When a session fails, see the log table for more information. Before you run the session again, drop the log table or enter a different table name in the session 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 session to load 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 session to load to Teradata.

The following table describes the session 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 session 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 PowerCenter Integration Service uses the name <code><error_database>.INFA_ET1_<number></code> . 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 PowerCenter Integration Service uses the name <code><error_database>.INFA_ET2_<number></code> . The exact table name appears in the session log.

When a session fails, see the error tables for more information about the errors. Before you run the session again, drop the error tables or enter different table names in the session 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 session to load 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 session to load to Teradata.

The following table describes the session 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 PowerCenter Integration Service uses the name <code><work_table_database>.INFA<number>_WT</code> . The exact table name appears in the session log.

Pipeline Partitioning

You can increase the number of partitions in a pipeline to improve session performance. Increasing the number of partitions allows the PowerCenter Integration Service to create multiple connections to sources and targets and process partitions of sources and targets concurrently.

You can configure partitioning for a session that extracts from or loads to Teradata. If the session loads to Teradata, set the system operator to **Stream**, **Load**, or **Update** in the connection object properties. If the sessions extracts from Teradata, set the system operator to **Export** in the connection object properties.

To update partitioning information, use the Partitions view on the Mapping tab of the session properties.

For a session that loads to Teradata, you can override session-level properties at the partition level for the first partition. Session-level properties of the first partition override the session-level properties of all the partitions in the session. You cannot provide different session-level properties for each partition.

The following table describes the partition types for partition points in Teradata PT API sessions:

Partition Point	Partition Type
Source Qualifier	Pass-through
Teradata PT API Target	Pass-through

Support for Multiple Instances

You can load data in parallel through multiple instances into a Teradata PT API target from a source to improve session performance. You can specify the number of parallel instances to load data into the Teradata PT API target as a session property. If you have multiple targets in a session, you can specify different number of instances for different targets. The session level properties are applicable for all the instances.

To achieve optimal session performance you can configure the number of partitions and number of instances based on factors like network latency, type of source, and frequency of data update.

Tip: For optimal performance, you can configure the number of partitions to be equal to the number of instances.

Recovery with Multiple Instances

You cannot recover sessions with multiple instances. Session fails if you enable recovery and run the session.

Staged Loading

Staged loading is the ability to pause an active load operation until additional data is available. Enable the session parameter pause acquisition to enable staged loading. When you run a session with pause acquisition enabled, the session extracts data from the source and pauses before it loads to the target. You can run the session multiple times to acquire data from different sources or different data from the same source. Disable pause acquisition and run the session to load the acquired data to the target at once.

For example, you want to load data into the same target table from two sources. Run the session with pause acquisition when you load data from the first source. Session pauses the load operation after acquiring data from the source. Disable pause acquisition and run the session with the second source. The session loads data from both the sources into the target table at once.

You cannot access the target until the session loads data from all the sources. You can use staged loading with Load and Update operators. If you enable pause acquisition, use the same operator for the subsequent runs of the session.

After you successfully run a session with pause acquisition, turn off Drop log, work, and error tables and Truncate target tables for subsequent runs. Use the same log, work, and error tables for the subsequent runs.

If you update the same target table through different sessions and use pause acquisition, specify the log, work, and error table names manually. Use the same table names for all sessions for the same target table.

Spool Modes

You can configure a session 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 session properties for Spool Mode:

- **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 session fails with an error.

Use the Spool Mode session property to configure the spool mode Teradata PT API uses to extract data from Teradata.

Configuring a Session for a Teradata PT API Source

You can configure a session to extract from Teradata. When you run a Teradata PT API session, you cannot use stored procedures, pushdown optimization, or row error logging. The PowerCenter Integration Service ignores source and source qualifier properties that you override in the session.

1. Change the reader type to Teradata Parallel Transporter Reader in the Readers settings on the Mapping tab.
2. From the Connections settings on the Sources node, select a Teradata PT connection.
3. From the Connections settings on the Sources node of the Mapping tab, configure the Teradata PT API source properties to enable Teradata PT API tracing. The following table describes the Teradata PT API source properties:

Property	Description
Spool Mode	<p>Determines the spool mode Teradata PT API uses to extract data from Teradata. You can choose one of the following spool modes:</p> <ul style="list-style-type: none">• 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. <p>Default is Spool.</p>

Property	Description
Driver Tracing Level	<p>Determines Teradata PT API tracing at the driver level:</p> <ul style="list-style-type: none"> • 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. <p>Default is TD_OFF.</p>
Infrastructure Tracing Level	<p>Determines Teradata PT API tracing at the infrastructure level:</p> <ul style="list-style-type: none"> • 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. <p>Default is TD_OFF.</p> <p>You must enable the driver tracing level before you can enable the infrastructure tracing level.</p>
Trace File Name	<p>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.</p>

Note: You can view load statistics in the session log. The load summary in the Workflow Monitor does not display load statistics.

Configuring a Session for a Teradata PT API Target

You can configure a session to load to Teradata. A Teradata PT API session cannot use stored procedures, pushdown optimization, or row error logging. The PowerCenter Integration Service ignores target properties that you override in the session.

The Workflow Manager allows you to create up to two connections for each target instance. The first connection defines the connection to Teradata PT API. The second connection defines an optional ODBC connection to the target database. Create a target ODBC connection when you enable the session or workflow for recovery, and you do not create the recovery table in the target database manually.

Select a Teradata target ODBC connection as the second connection for the target instance if you want to perform any of the following actions:

- Enable the session or workflow for recovery without creating the recovery table in the target database manually.
- Drop log, error, and work tables.
- Truncate target tables.

Otherwise, leave the second connection empty.

Note: If you want to run an update or delete operation on a Teradata target table that does not have a primary key column, you must edit the target definition and specify at least one connected column as a primary key column.

To configure a session to load to Teradata:

1. Change the writer type to Teradata Parallel Transporter Writer in the Writers settings on the Mapping tab.
2. From the Connections settings on the Targets node, select a Teradata PT connection.
3. From the Connections settings on the Targets node of the Mapping tab, configure the Teradata PT API target properties. The following table describes the Teradata PT API target properties:

Property	Description
Work Table Database	Name of the database that stores the work tables.
Work Table Name	Name of the work table.
Macro Database	<p>Name of the database that stores the macros Teradata PT API creates when you select the Stream system operator.</p> <p>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.</p> <p>If you do not specify a macro database, Teradata PT API stores the macros in the log database.</p>
Pause Acquisition	<p>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.</p> <p>Default is disabled.</p>
Instances	<p>The number of parallel instances to load data into the Teradata PT API target.</p> <p>Default is 1.</p>
Query Band Expression	<p>The query band expression to be passed to the Teradata PT API.</p> <p>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, <code>ApplicationName=Informatica;Version=9.0.1;ClientUser=A;</code></p>
Insert	<p>Teradata PT API creates a DML group to insert rows.</p> <p>If you do not want to insert rows, clear this option to increase session performance.</p> <p>Default is selected.</p>
Update	<p>Teradata PT API creates a DML group to update rows.</p> <p>If you do not want to update rows, clear this option to increase session performance.</p> <p>Default is selected.</p>
Delete	<p>Teradata PT API creates a DML group to delete rows.</p> <p>If you do not want to delete rows, clear this option to increase session performance.</p> <p>Default is selected.</p>

Property	Description
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	<p>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.</p> <p>Use this attribute in conjunction with the Replacement Character attribute. The PowerCenter Integration Service ignores this attribute if you did not specify a replacement character while loading data to Teradata targets.</p> <p>Default is 8x-13x.</p>
Update Else Insert	<p>Teradata PT API updates existing rows and inserts other rows as if marked for update. If disabled, Teradata PT API updates existing rows only.</p> <p>The PowerCenter Integration Service ignores this attribute when you treat source rows as inserts or deletes.</p> <p>Default is disabled.</p>
Truncate Table	<p>Teradata PT API deletes all rows in the Teradata target before it loads data.</p> <p>This attribute is available for the Update and Stream system operators. It is available for the Load system operator if you select a Teradata target ODBC connection.</p> <p>Default is disabled.</p>
Mark Missing Rows	<p>Specifies how Teradata PT API handles rows that do not exist in the target table:</p> <ul style="list-style-type: none"> - 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. <p>Default is None.</p>
Mark Duplicate Rows	<p>This attribute is available for the Update and Stream system operators.</p> <p>Specifies how Teradata PT API handles duplicate rows when it attempts to insert or update rows in the target table:</p> <ul style="list-style-type: none"> - 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. <p>Default is For Insert.</p>

Property	Description
Mark Extra Rows	<p>Specifies how Teradata PT API marks error rows when it attempts to update or delete multiple rows in the target table:</p> <ul style="list-style-type: none"> - 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. <p>Default is Both.</p>
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 TableName2	Name of the second error table.
Drop Log/Error/Work Tables	<p>Drops existing log, error, and work tables for a session when the session starts. This attribute is available if you select a Teradata target ODBC connection. Default is disabled.</p>
Serialize	<p>Uses the Teradata PT API serialize mechanism to reduce locking overhead when you select the Stream system operator. Default is enabled.</p>
Serialize Columns	<p>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.</p>
Pack	<p>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.</p>
Pack Maximum	<p>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.</p>

Property	Description
Buffers	<p>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:</p> $\text{Max_Request_Buffers} = \text{Buffers} * \text{Number_Connected_Sessions}$ <p>Must be a positive, nonzero integer. Default is 3. Minimum is 2.</p>
Error Limit	<p>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.</p>
Replication Override	<p>Specifies how Teradata PT API overrides the normal replication services controls for an active Teradata PT API session:</p> <ul style="list-style-type: none"> - 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. <p>Default is None.</p>
Driver Tracing Level	<p>Determines Teradata PT API tracing at the driver level:</p> <ul style="list-style-type: none"> - 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. <p>Default is TD_OFF.</p>
Infrastructure Tracing Level	<p>Determines Teradata PT API tracing at the infrastructure level:</p> <ul style="list-style-type: none"> - 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. <p>Default is TD_OFF.</p> <p>You must enable the driver tracing level before you can enable the infrastructure tracing level.</p>

Property	Description
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.
Write Buffer Size	<p>This property is applicable when you use the Load system operator.</p> <p>Defines the maximum buffer size in kilobytes that the Teradata PT API job uses for writing data.</p> <p>If you do not enter a value and you use TTU version 15.10 or later, the PowerCenter Integration Service sets the value as 1024 KB for optimal performance.</p>

APPENDIX A

Datatype Reference

This appendix includes the following topics:

- [Datatype Reference Overview, 25](#)
- [Teradata and Transformation Datatypes, 25](#)

Datatype Reference Overview

PowerCenter uses the following datatypes in Teradata mappings:

- Teradata native datatypes. Teradata datatypes appear in Teradata definitions in a mapping.
- Transformation datatypes. Set of datatypes that appear in the transformations. They are internal datatypes based on ANSI SQL-92 generic datatypes, which the PowerCenter Integration Service uses to move data across platforms. They appear in all transformations in a mapping.

When the PowerCenter Integration Service reads source data, it converts the native datatypes to the comparable transformation datatypes before transforming the data. When the PowerCenter Integration Service writes to target, it converts the transformation datatypes to the comparable native datatypes.

Teradata and Transformation Datatypes

The following table compares Teradata datatypes and transformation datatypes:

Teradata	Range	Transformation	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	Small Integer	Precision 5, scale 0
Char	1 to 64,000 bytes	String	1 to 104,857,600 characters

Teradata	Range	Transformation	Range
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
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	Small Integer	Precision 5, scale 0
Time	00:00:00.000000 to 23:59:61.999999 Precision 8, scale 0	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

Datatype Synonyms

The following table compares Teradata synonyms to transformation datatypes:

Synonym	Transformation
Double Precision	Double
Numeric	Decimal
Real	Double

INDEX

C

connections
Teradata PT API [11](#)

D

datatypes
Teradata [25](#)

E

environment variables
Teradata PT API, setting [9](#)
error tables
Teradata PT, description [15](#)
Teradata PT, dropping [19](#)

L

log tables
Teradata PT API, description [15](#)
Teradata PT, dropping [19](#)

M

macro database
Teradata PT [19](#)

P

partitioning
Teradata PT API sessions [16](#)
PowerExchange for Teradata connections
properties [12](#)
PowerExchange for Teradata Parallel Transporter API
configuring [8](#)

PowerExchange for Teradata Parallel Transporter API (*continued*)
upgrading [8](#)
prerequisites
PowerExchange for Teradata Parallel Transporter API [8](#)

R

recovery
Teradata PT API sessions [14](#)
recovery tables
Teradata PT API, creating automatically [14](#)
Teradata PT, creating manually [14](#)

S

sessions
configuring to extract from Teradata PT [18](#)
configuring to load to Teradata PT [19](#)
synonym datatypes
Teradata [26](#)

T

targets
Teradata PT, primary key columns [19](#)
Teradata
comparing datatypes to transformation [25](#)
datatypes [25](#)
Teradata PT API connections
creating [11](#)
transformation datatypes
comparing to Teradata [25](#)

W

work tables
Teradata PT API, description [16](#)
Teradata PT, dropping [19](#)