Microsoft Connector for Teradata by Attunity

SQL Server Technical Article

Writer: Doug Wheaton (Attunity)

Technical Reviewers: Ramakrishnan Krishnan (Microsoft), Rupal Shah (Teradata)

Published: November 2013

Applies to: SQL Server 2012 and 2008

Summary: This white paper presents detailed information on Microsoft Connector for Teradata by Attunity. It contains an overview of the architecture, installation and configuration information, common use cases, and detailed technical appendices.

Copyright

Contents

Introduction 5

Architecture 7

Prerequisites and Installation 9

- SQL Server Components 9
- Required Teradata Components 10
- Installation 11

General Configuration in SQL Server Business Intelligence Development Studio 15

- Enabling Teradata Source and Teradata Destination as Data Flow Toolbox Items 15
- Configuring the Teradata Connection Manager 17

Use Case 1: Bulk Unload from Teradata to SQL Server Using TPT Export 19

- Use Case Architecture Overview 19
- Description 19
- Defining the SQL Server Integration Services Data Flow 19
  - Configuring the Teradata Source Component 19
  - Choosing the Data Access Mode 23
  - Adding and Configuring the Destination Component 23

Use Case 2: Bulk Load into Teradata Using TPT Load 24

- Use Case 2 Architecture Overview 24
- Description 24
- Defining the SQL Server Integration Services Data Flow 24
  - Adding and Configuring Source and Teradata Destination Components 24

Use Case 3: Incremental Load into Teradata Using TPT Stream 26

- Description 26
- Architecture 27
- Defining the SQL Server Integration Services Data Flow 27
  - Adding and Configuring Source and Teradata Destination Components 27

Conclusion 29

Appendix A – Data Types 31

- Supported Data Types 31
- Data Type Mapping 31

Appendix B – Teradata Source Advanced Topics 35

Teradata Source..35
Error Handling..35
Parallelism 35
Troubleshooting the Teradata Source 35
Configuring the Teradata Source..36
Teradata Source Editor (Connection Manager Page)..36
Teradata Source Editor (Columns Page) 37
Teradata Source Editor (Error Output Page) 39
Teradata Source Advanced Editor 40
Teradata Source Custom Properties..41
Appendix C – Teradata Destination Component Advanced Topics 45
Teradata Destination 45
Load Options 45
Error Handling..45
Parallelism 47
Troubleshooting the Teradata Destination..47
Configuring the Teradata Destination 47
Teradata Destination Editor (Connection Manager Page) 47
Teradata Destination Editor (Mappings Page)..50
Teradata Destination Editor (Error Output Page)..51
Teradata Destination Advanced Editor..52
Teradata Destination Custom Properties 53
Appendix D – Troubleshooting Run-Time Failures..57
STEP 1– Eliminate the Common Problems 57
STEP 2 - Using the Logging Facility 59
STEP 3 - Understanding the Log File 61

Introduction

The Microsoft® Connector for Teradata by Attunity provides a high-performance means of loading and unloading data to and from Teradata databases. It is available as a free download from the Microsoft and Attunity Web sites for SQL Server Enterprise and Developer edition customers. It enables the high-volume data movement to and from Teradata within the Microsoft SQL Server® environment via seamless interfaces in both Full and Incremental modes. The Microsoft Connector for Teradata by Attunity integrates the Teradata Parallel Transporter (sometimes abbreviated as TPT) Application Programming Interface (API) and the TPT Load, Stream, and Export operators.

Attunity, a Microsoft OEM partner has produced several connectors for SQL Server Integration Services in the past. The following table summarizes the supported configurations with Integration Services running SQL Server 2008 and 2012 on Enterprise and Developer versions. This connector is supported by Microsoft.

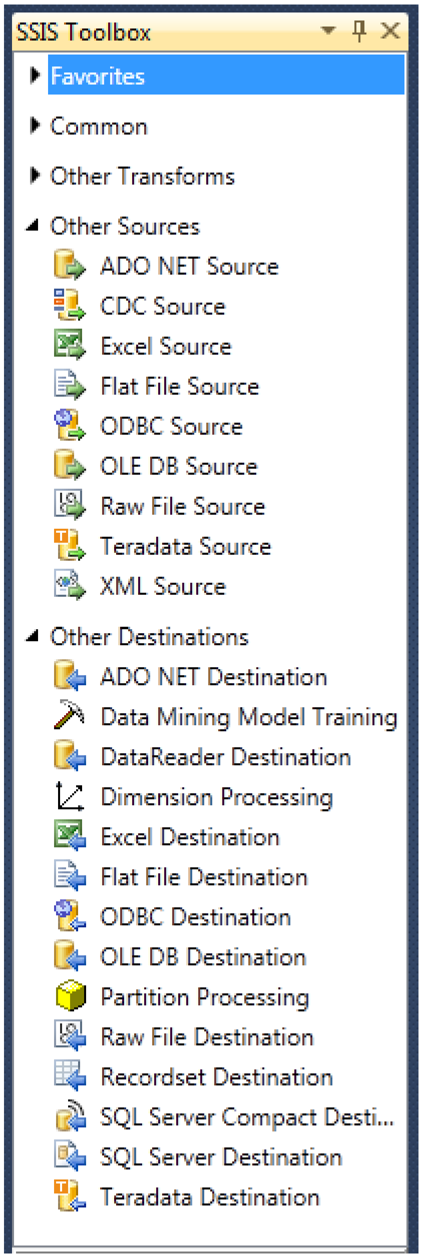
Teradata Database	14.0	13.10	13.0	12.0	6.2 – 6.0
Microsoft Connector 1.0 (SQL Server 2008)	√	√	√	√	√
Microsoft Connector 2.0 (SQL Server 2012)	√	√	√	√	√

Table 1: Support Matrix

Teradata Parallel Transporter API is a set of C++ application programming interfaces used to load and unload data to and from Teradata systems. Teradata Parallel Transporter API, with the TPT operators (Load, Stream, and Export), enables an application to access Teradata Database using proprietary Teradata load and unload protocols. The Teradata Parallel Transporter API is a functional library that is part of the Microsoft Connector for Teradata by Attunity, and it provides the SQL Server Integration Services application with more control during the load and unload processes.

While the Microsoft Connector for Teradata by Attunity offers an easy and streamlined user experience, this simplicity cannot come without understanding of Teradata Database and without the necessary permissions for management operations such as table deletion, table creation, and so on.

The Microsoft Connector for Teradata is co-developed by Attunity partnership with SQL Server Integration Services Development Team, and it utilizes the Teradata Parallel Transporter API and internal buffering with Teradata PT operators directly rather than requiring access through a managed interface. As a result, it offers a significant performance advantage over other connectors. The Microsoft Connector for Teradata by Attunity has been designed to look and function like part of SQL Server Data Tools (SSDT) designer in SQL Server 2012 (former called Business Intelligence Development Studio (BIDS) in SQL Server 2008); the connector handles changes in a manner that is consistent with built-in connectors, and it captures and reports errors generated by the source systems to which it connects. After you install the Microsoft Connector for Teradata, you can create connection managers that use the Connector in the usual way. New data flow sources and data flow destinations that reference these connection managers are added to the data flow toolbox in SQL Server Data Tools (Figures 1 and 2).



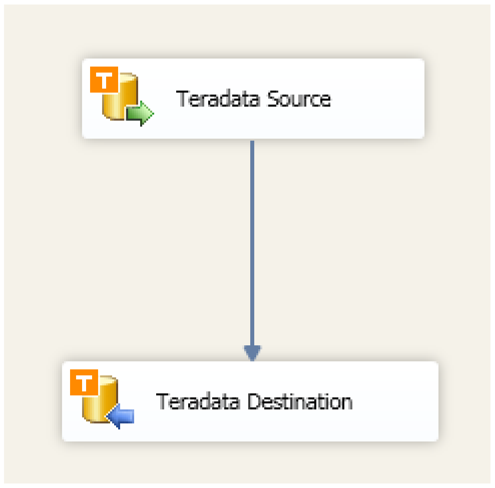


Figure 1: Data Flow Toolbox

Figure 2: Teradata components

A **Teradata Source** and **Teradata Destination** are demonstrated as shown in Figure 2 as part of the data flow.

The Microsoft Connector for Teradata is compatible with Teradata Database versions as shown in Table 1 above and supports SQL Server Integration Services on x86 and x64 platforms. As a data source, the Microsoft Connector for Teradata uses the Teradata PT Export operator to unload data from tables or views. It can also use an SQL statement. The data destination can load data into Teradata databases by using incremental loading, using the Teradata PT Stream operator, or by fast loading an empty table, using the Teradata PT Load operator.

The Microsoft Connector for Teradata by Attunity has three main components:

- Teradata Source, to unload data in bulk from Teradata
- Teradata Destination, to load data in bulk or incrementally into Teradata
- Teradata Connection Manager, to enable a package to connect to a Teradata data source

Architecture

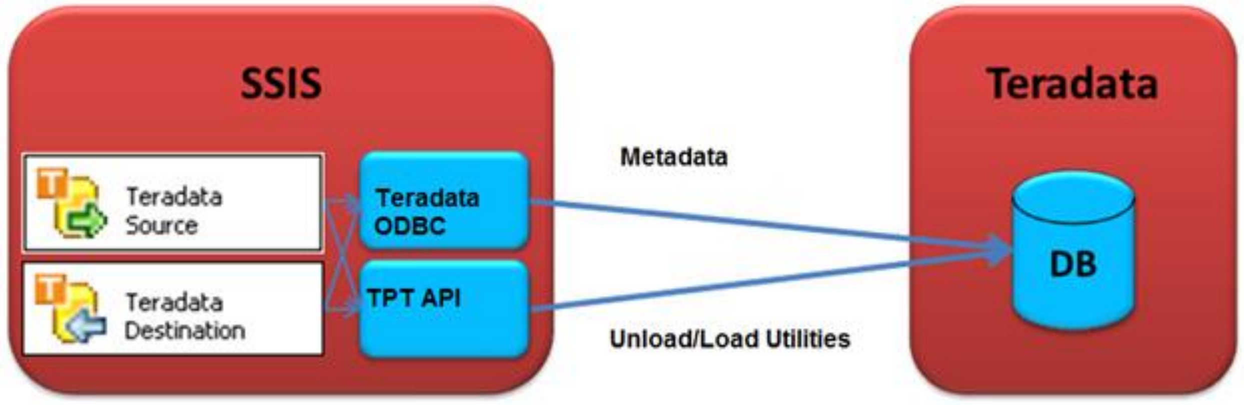


Figure 3: General Teradata Connector architecture components for SQL Server Integration Services (SSIS)

The following Teradata PT operators are used to achieve optimal performance:

- Teradata PT Load operator - Bulk load to an empty table
- Teradata PT Stream operator - Incremental load to an existing table
- Teradata PT Export operator – Unload a table or SQL command

Because Teradata Parallel Transporter does not support metadata retrieval, all metadata access operations (including dynamic description of SQL statements) are done using the ODBC Driver for Teradata (available on the X86 and X64 Windows® platforms).

Below are past and current enhancements for Microsoft Connector for Teradata.

Microsoft Connector Version 1.1 for Teradata, by Attunity

Performance Improvements for Teradata Components

- Optimize conversion functions in the Teradata Source and Destination components.

Additional Property Support for the Teradata Components

Added support for the following properties in the Teradata Destination:

- Robust: TPT Stream only
- ArraySupport: TPT Stream only
- Buffers: TPT Stream only
- BufferSize: TPT Load only
- QueryBandSessionInfo
- DetailedTracingLevel

Added support for the following properties in the Teradata Source:

- BufferSize
- QueryBandSessionInfo
- DetailedTracingLevel

Support for TPT API Version 13 in the Teradata Components
Support was added for TPT API version 13 Edition 2 (13.0.0.2). Version 12 APIs are still supported.

Enhanced Logging for the Teradata components
The DetailedTraceLevel property was added to the Teradata source and destination. This allows setting the TPT API tracing to different levels.

Query Banding is supported by the Teradata components
The Teradata source and destination support query banding. This allows charge back, monitoring, and governance. This is set by the new QueryBandSessionInfo property

Microsoft Connector Version 1.2 for Teradata by Attunity

The components are written to achieve optimal performance when loading data into Teradata or unloading data from Teradata in the context of Microsoft SSIS.

Microsoft SSIS Connectors by Attunity Version 1.2 is a minor release. It supports SQL Server 2008 Integration Services and includes performance enhancements, bug fixes, and continued improvements for ease of use.

The following additional enhancements were made:
- Teradata Enhancement - Support for TPT API 14.0 and TPT API 13.10.

- Support was added for Teradata 13.10 and Teradata 14.0 by supporting TPT API version 13.0 and TPT API version 14.0 (14.00.00.02 or higher). Previous versions of the TPT API are still supported.
- Teradata Enhancement - Starting from Teradata 13.0 Teradata database supports DDL with no primary index which was introduced specifically for loading data faster into Teradata; these tables were not listed in the table dropdown list.
- Teradata Bug Fix - Teradata DDL supports column names being reserved words. Teradata Connector could not work with tables having such columns.
- Teradata Bug Fix - Teradata destination component failed to read problematic rows from the error table with the following error: “Errors while trying to read error tables. Invalid TPT operator.”

Microsoft Connector Version 2.0 for Teradata by Attunity

The components are written to achieve optimal performance when loading data into Teradata or unloading data from Teradata in the context of Microsoft SSIS.

Microsoft SSIS Connectors by Attunity Version 2.0 is a minor release. It supports SQL Server 2012 Integration Services and includes bug fixes and support for updated Teradata product releases.

The following additional enhancements were made:
- Teradata Bug Fixes – Some hot key issues were fixed in the Teradata connectors UI.
- Teradata Bug Fixes – Localization issues were fixed in the Teradata connectors.
- Teradata Enhancement - Support for TPT API 14.0 and TPT API 13.10.
- Support was added for Teradata 13.10 and Teradata 14.0 by supporting TPT API version 13.0 and TPT API version 14.0 (14.00.00.02 or higher). Previous versions of the TPT API are still supported.
- Teradata Enhancement - Starting from Teradata 13.0 Teradata database supports DDL with no primary index which was introduced specifically for loading data faster into Teradata; these tables were not listed in the table dropdown list.
- Teradata Bug Fix - Teradata DDL supports column names being reserved words. Teradata Connector could not work with tables having such columns.
- Teradata Bug Fix - Teradata destination component failed to read problematic rows from the error table with the following error: “Errors while trying to read error tables. Invalid TPT operator”.

Prerequisites and Installation

SQL Server Components

The following Microsoft SQL Server products are supported by the SQL Server Integration Services components for Teradata:

- Microsoft SQL Server 2008 and 2012 Enterprise and Developer
- Microsoft Business Intelligence Development Studio (BIDS) for SQL Server 2008 and SQL Server Development Tools (SSDT) for SQL Server 2012

These versions are supported on the following operating systems and platforms:

- Windows 7
- Windows Vista® 32-bit (x86) and 64-bit (x64)
- Windows Server® 2003 32-bit (x86) and 64-bit (x64)*
- Windows Server 2008 32-bit (x86) and 64-bit (x64)

(*) Not applicable for Microsoft Connector 2.0 for Teradata, by Attunity

Required Teradata Components

We highly encourage customers to work with Teradata Professional Services when installing Teradata Parallel Transporter or any Teradata Tools and Utilities products. At a high level, the following Teradata client products need to be installed. The client products have both 32-bit and 64-bit component offerings, depending on customer platform requirements. It is 'highly' recommended to install 32-bit and 64-bit on a 64-bit platform:

- Teradata Parallel Transporter API and Operators
- ODBC Driver for Teradata

Previous versions of the TPT API are still supported by the Microsoft Connectors for Teradata. Specifically, for the Microsoft Connector for Teradata, you must install the Teradata Parallel Transporter API, the TPT operators, and the required dependencies listed below. You should install the products in the order listed below or how each product states its prerequisites during installation or in the manual. Installing from the TTU CD is recommended and will install dependencies on default and install 32-bit and 64-bit components on default when installing on a 64-bit platform. This is particularly important and required if you plan to use BIDS or SSDT (32-bit application) on the same SSIS platform for designing, testing and running your SSIS packages.

<u>Teradata Parallel Transporter API Software and Dependencies</u>

1. Teradata International Components for Unicode (Teradata ICU)
2. Teradata Generic Security Services (TeraGSS)
3. Teradata Call Level Interface Version 2 (Teradata CLIv2)
4. Teradata Parallel Transporter Base
5. Teradata Parallel Transporter Stream

Unlike previous versions of TPT version 14.0 and greater has bundled TPT API and Operators (i.e. Load and Export) into Teradata Parallel Transporter Base and Teradata Parallel Transporter Stream operator into its own separate install. Please review appropriate release documentation for TPT installation instructions if you require 13.10 or earlier TPT versions at your site.

Note, TPT releases are forward and backward compatible with the Teradata Database release. For example, TPT version 14.00 can work with Teradata Database release 14.10. Or TPT version 14.10 can work with Teradata Database 14.00. But, different or multiple versions of TPT releases cannot co-exist (not supported) on the same machine

In addition, install appropriate Microsoft Connector for Teradata based on SQL Server version (see table above).

For Teradata products, the full version number for a package is 14.10.00.xx. The last two numbers in the version string denote the e-fix version, with the latest release having the greatest e-fix number. Before you install any Teradata product, you should check with Teradata to see whether a greater e-fix release exists. Before you use an e-fix version of a product, review the corresponding e-fix *ReadMe* file for fix information and for documentation on any resulting usage changes.

On 32-bit or 64-bit computers, install the following:

- Teradata Parallel Transport Base and Stream software and dependencies*
- ODBC Driver for Teradata

(*) Dependencies mentioned above install on default (i.e. not visible during installation) with versions 14.00 and higher.

In the case, where SSIS packages are developed on a 32-bit system, but will be execute on a SSIS 64-bit system. Remember to set the 'Run64BitRuntime' project property accordingly.

Note: To run the package in SQL Server Data Tools or Business Intelligence Development Studio, you must configure the project to run in 32-bit mode.

In the project properties for a SQL Server Integration Services package, you can select 32-bit execution by setting the value of the **Run64BitRuntime** property on the **Debugging** page to **False**. By default, the value of this property is **True**. When the 64-bit version of the SQL Server Integration Services runtime is not installed, this setting is ignored.

Installation

The Microsoft Connector for Teradata by Attunity is available as a Web download. It can be downloaded from the Microsoft Web site http://www.microsoft.com/en-us/download/default.aspx search for 'Microsoft Connector for Teradata' or for specific release:

For SQL Server 2008 use Microsoft Connector 1.2 for Teradata, by Attunity

http://www.microsoft.com/en-us/download/details.aspx?id=29284

For SQL Server 2012 use Microsoft Connector 2.0 for Teradata, by Attunity

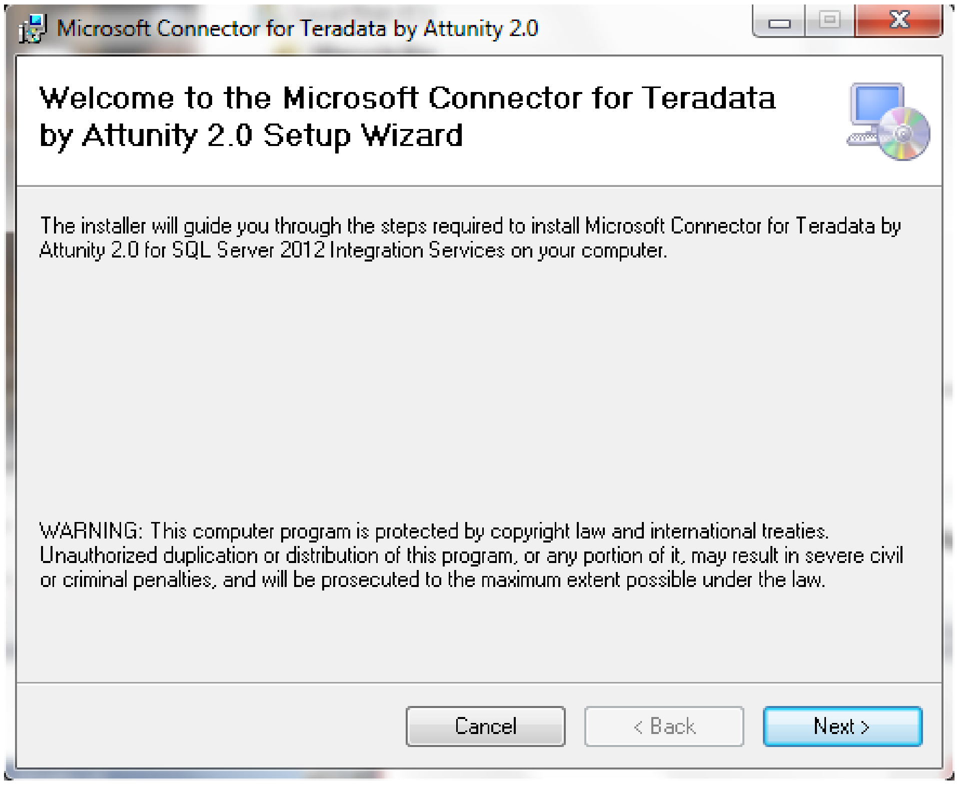
http://www.microsoft.com/en-us/download/details.aspx?id=29283

Take care to download the appropriate installation kit and platform requirements for your needs (i.e. 32-bit, 64-bit or both).

These general steps apply to all of the use cases described later:

1. Install the Microsoft Connector for Teradata by Attunity.
2. Enable Teradata Source and Teradata Destination as data flow sources.
3. Set up the Connection Manager for Teradata.

To install, simply double-click the installation package.

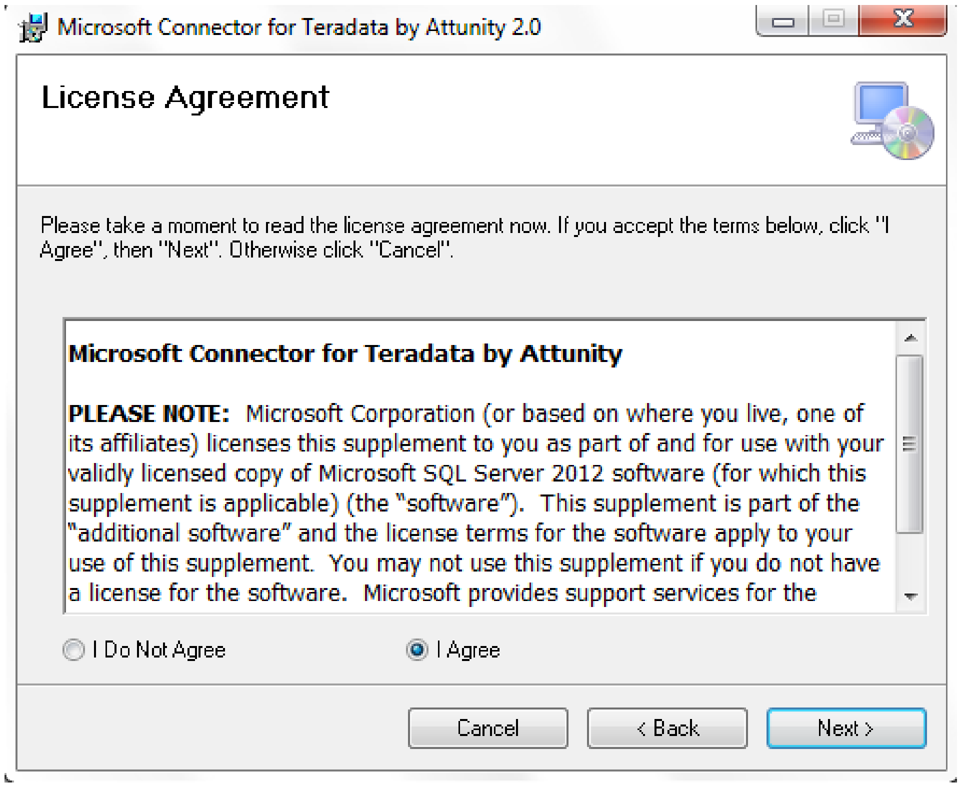

Microsoft Connector for Teradata by Attunity 2.0

Welcome to the Microsoft Connector for Teradata by Attunity 2.0 Setup Wizard

The installer will guide you through the steps required to install Microsoft Connector for Teradata by Attunity 2.0 for SQL Server 2012 Integration Services on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure4: Installation process


Microsoft Connector for Teradata by Attunity 2.0

License Agreement

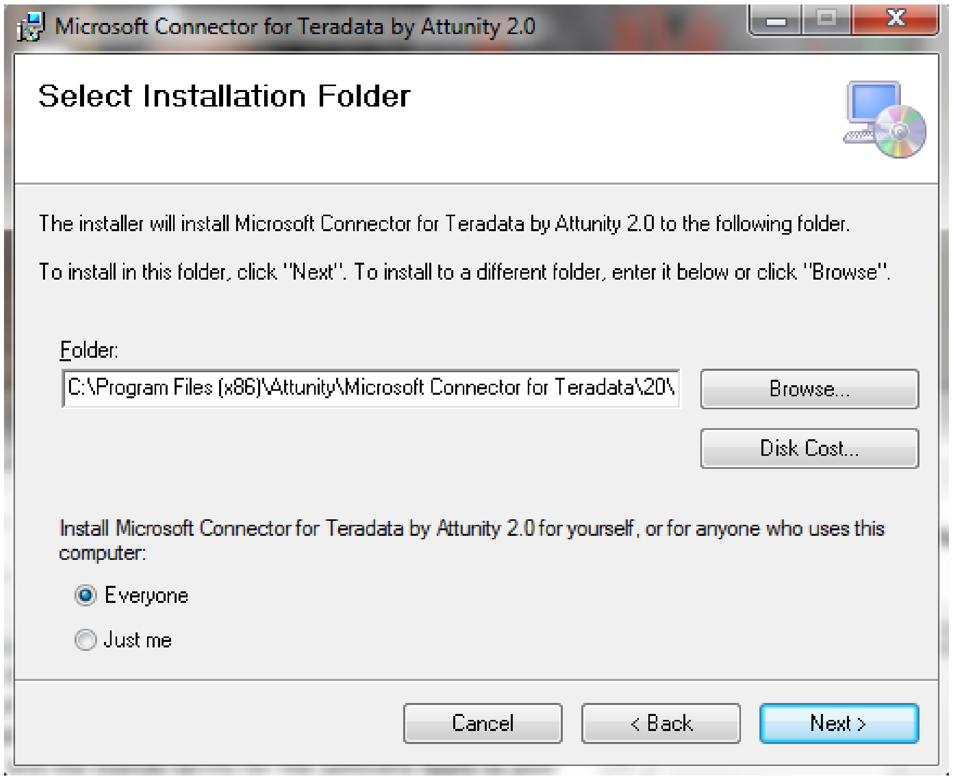
Please take a moment to read the license agreement now. If you accept the terms below, click "I Agree", then "Next". Otherwise click "Cancel".

Microsoft Connector for Teradata by Attunity

PLEASE NOTE: Microsoft Corporation (or based on where you live, one of its affiliates) licenses this supplement to you as part of and for use with your validly licensed copy of Microsoft SQL Server 2012 software (for which this supplement is applicable) (the "software"). This supplement is part of the "additional software" and the license terms for the software apply to your use of this supplement. You may not use this supplement if you do not have a license for the software. Microsoft provides support services for the

I Do Not Agree [✓ I Agree]

Figure 5: Installation process


Microsoft Connector for Teradata by Attunity 2.0

Select Installation Folder

The installer will install Microsoft Connector for Teradata by Attunity 2.0 to the following folder.

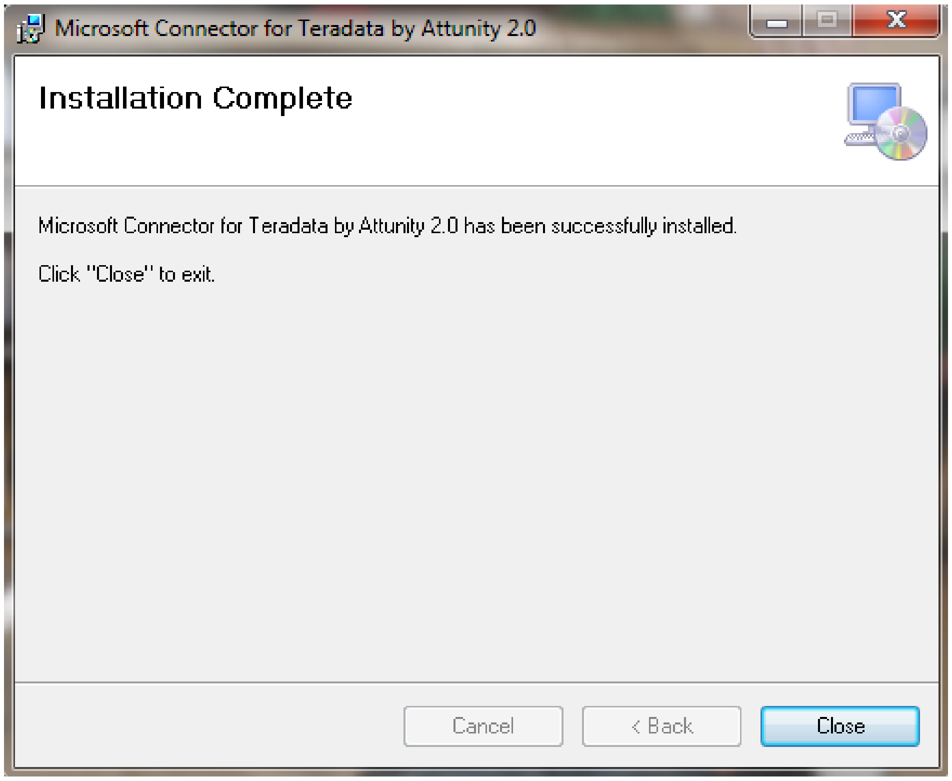
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\Program Files (x86)\Attunity\Microsoft Connector for Teradata\20\

Install Microsoft Connector for Teradata by Attunity 2.0 for yourself, or for anyone who uses this computer:

[✓ Everyone] Just me

Figure 6: Installation process


Microsoft Connector for Teradata by Attunity 2.0

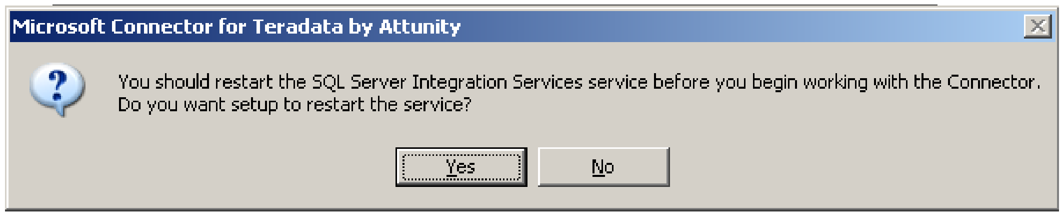
Installation Complete

Microsoft Connector for Teradata by Attunity 2.0 has been successfully installed.

Click "Close" to exit.

Figure 7: Installation process

After installation is complete, you will need to restart SQL Server Integration Services


Microsoft Connector for Teradata by Attunity

You should restart the SQL Server Integration Services service before you begin working with the Connector.
Do you want setup to restart the service?

Figure 9: Installation process

Figure 10 shows the menu path for accessing the connector.

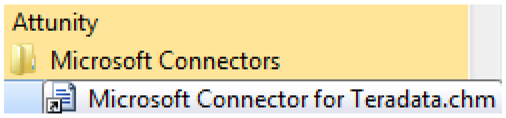


Figure 10: Programs menu

General Configuration

This section contains information about configuring the connector after it is installed.

Enabling Teradata Source and Teradata Destination as Data Flow Toolbox Items

In SQL Server 2008, after you install the connector, go to Business Intelligence Development Studio and create a new Integration Services project. The Microsoft Connector for Teradata components are not enabled by default as data flow toolbox items. To enable them on the **Tools** menu, click **Choose Toolbox Items**.

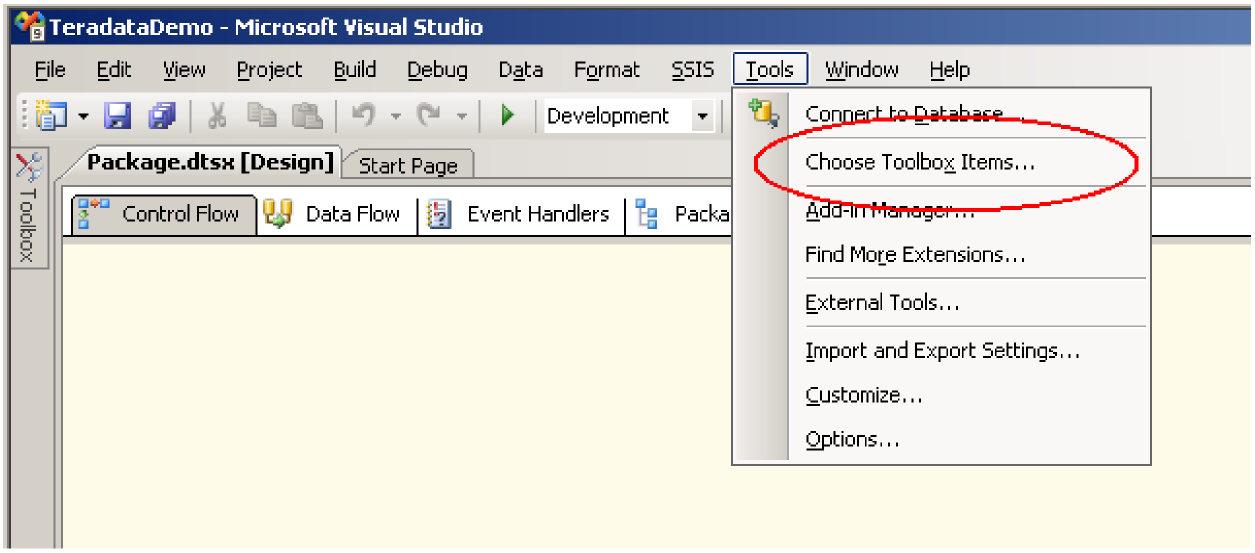
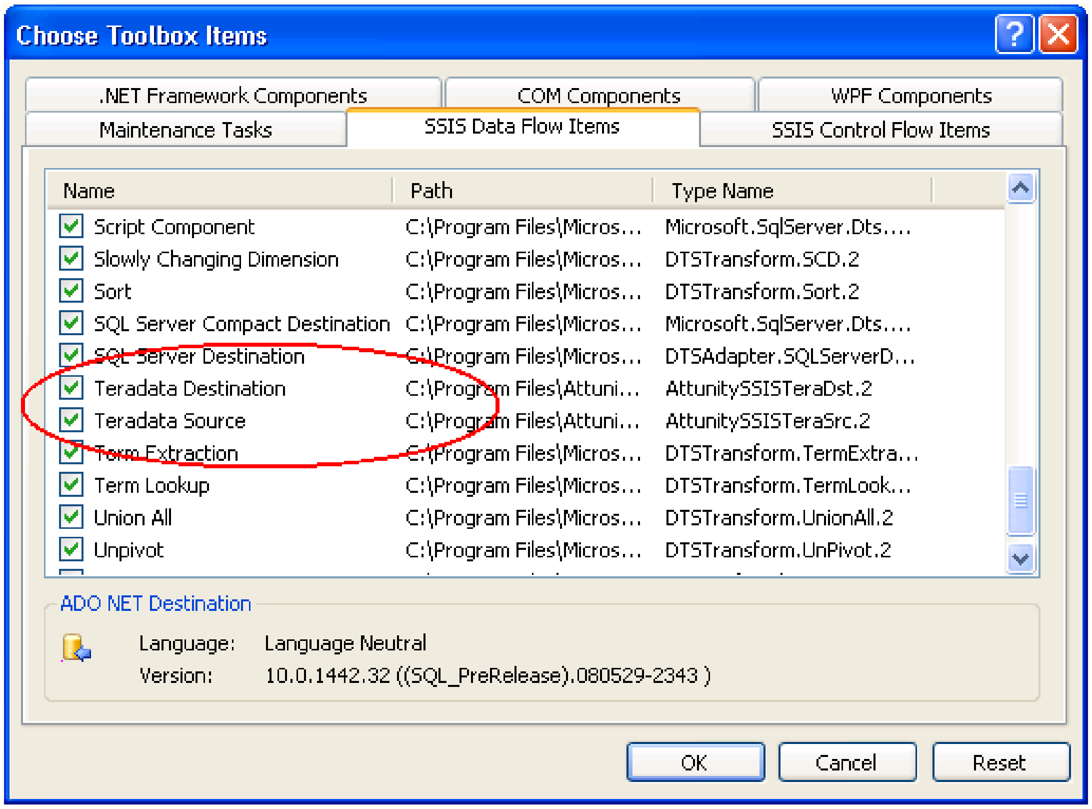


Figure 11: Choose Toolbox items

Then, on the **SSIS Data Flow Items** tab, select the **Teradata Source** and **Teradata Destination** check boxes.



Name	Path	Type Name
✓ Script Component	C:\Program Files\Micros...	Microsoft.SqlServer.Dts....
✓ Slowly Changing Dimension	C:\Program Files\Micros...	DTSTransform.SCD.2
✓ Sort	C:\Program Files\Micros...	DTSTransform.Sort.2
✓ SQL Server Compact Destination	C:\Program Files\Micros...	Microsoft.SqlServer.Dts....
✓ SQL Server Destination	C:\Program Files\Micros...	DTSAdapter.SQLServerD...
✓ Teradata Destination	C:\Program Files\Attuni...	AttunitySSISTeraDst.2
✓ Teradata Source	C:\Program Files\Attuni...	AttunitySSISTeraSrc.2
✓ Term Extraction	C:\Program Files\Micros...	DTSTransform.TermExtra...
✓ Term Lookup	C:\Program Files\Micros...	DTSTransform.TermLook...
✓ Union All	C:\Program Files\Micros...	DTSTransform.UnionAll.2
✓ Unpivot	C:\Program Files\Micros...	DTSTransform.UnPivot.2

ADO NET Destination

Language: Language Neutral
Version: 10.0.1442.32 ((SQL_PreRelease).080529-2343)

Figure 12: Selecting Teradata components

Now **Teradata Source** and **Teradata Destination** are available in the Data Flow Toolbox (Figure 13).

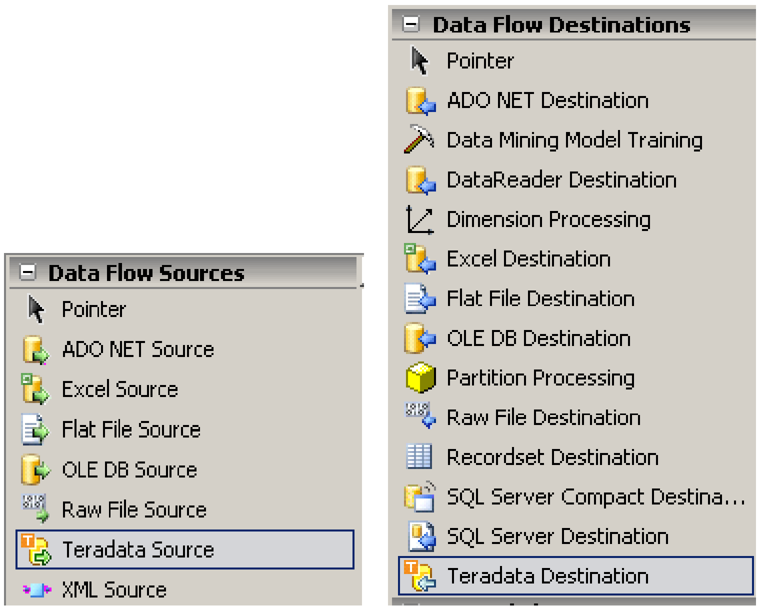
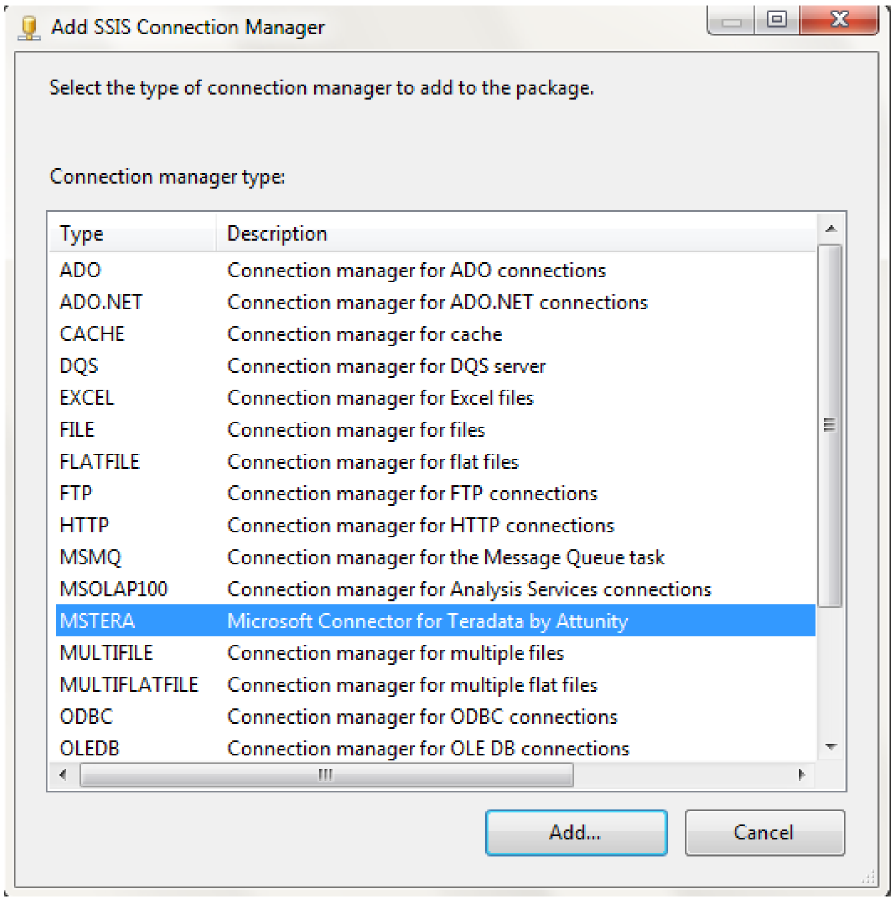


Figure 13: Teradata components in Toolbox

Note, in SQL Server 2012, after installing the connector components are enabled by default as data flow toolbox items. Though, sometimes may appear under Common versus Source and Destination folders. User can right click and move to appropriate folder.

Configuring the Teradata Connection Manager

In the SQL Server Integration Services project, add a new connection. Choose MSTERA as the connection manager type.



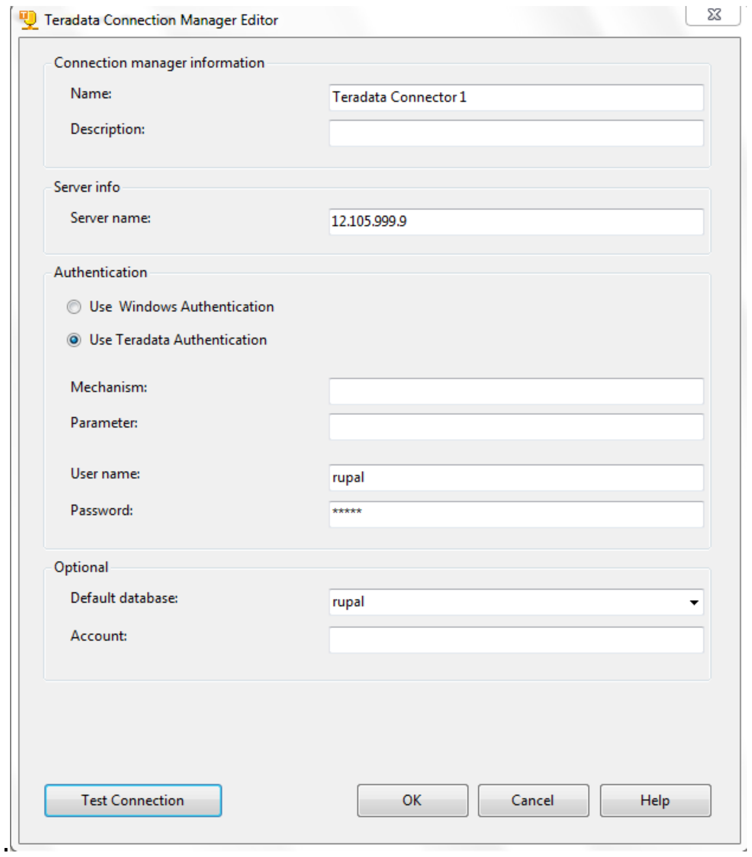
Select the type of connection manager to add to the package.

Connection manager type:

Type	Description
ADO	Connection manager for ADO connections
ADO.NET	Connection manager for ADO.NET connections
CACHE	Connection manager for cache
DQS	Connection manager for DQS server
EXCEL	Connection manager for Excel files
FILE	Connection manager for files
FLATFILE	Connection manager for flat files
FTP	Connection manager for FTP connections
HTTP	Connection manager for HTTP connections
MSMQ	Connection manager for the Message Queue task
MSOLAP100	Connection manager for Analysis Services connections
MSTERA	Microsoft Connector for Teradata by Attunity
MULTIFILE	Connection manager for multiple files
MULTIFLATFILE	Connection manager for multiple flat files
ODBC	Connection manager for ODBC connections
OLEDB	Connection manager for OLE DB connections

Figure 14: Connection manager type

After the connection is created, edit it, enter the server and logon information, and then click **Test Connection**.



Teradata Connection Manager Editor

Connection manager information

Name:	Teradata Connector 1
Description:	

Server info

Server name:	12.105.999.9

Authentication

( ) Use Windows Authentication
[✓ Use Teradata Authentication]

Mechanism:	
Parameter:	
User name:	rupal
Password:	*****

Optional

Default database:	rupal
Account:	

Figure 15: Teradata Connection Manager Editor

- The connection parameters are the same as you use with other Teradata tools.
- The connection manager is used by the Source and Destination components.
- The Teradata connection manager uses the ODBC Driver for Teradata to connect to the back-end Teradata source.

Use Case 1: Bulk Unload from Teradata to SQL Server Using TPT Export

This section describes a specific use cases for the connector. It begins with an overview of the architecture and steps through configuration.

Use Case Architecture Overview

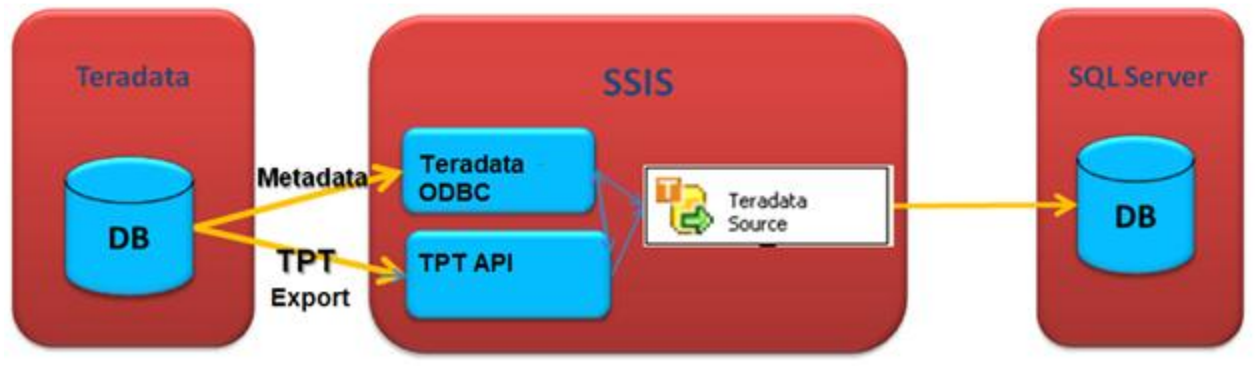


Figure 16: Use Case 1 architecture

Description

The challenge of unloading large amounts of data can be solved by using the Microsoft Connector for Teradata by Attunity component in Integration Services.

- Source table: in Teradata
- Target table: in SQL Server
- Data flow:
    - Configure Teradata Source Component
    - Configure SQL Server Destination Component

Defining the SQL Server Integration Services Data Flow

Configuring the Teradata Source Component

Add the component to the SQL Server Integration Services package data flow.

Select the Teradata Source component from the Data Flow toolbox.

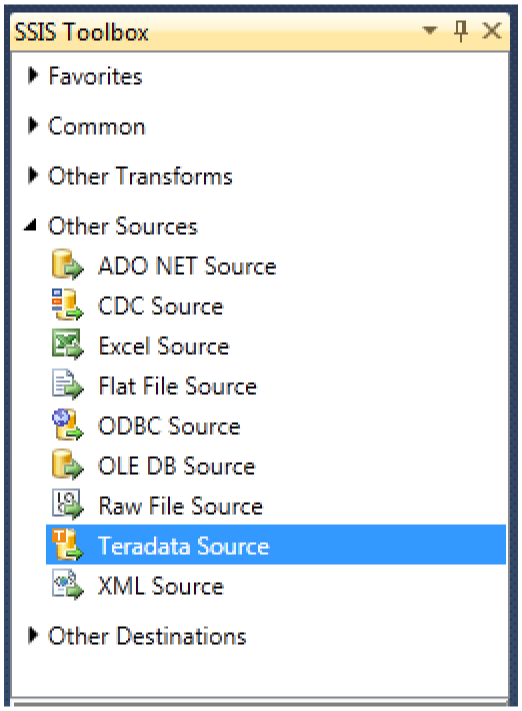


Figure17: Selecting the data source

Figure 18 shows the source and destination.

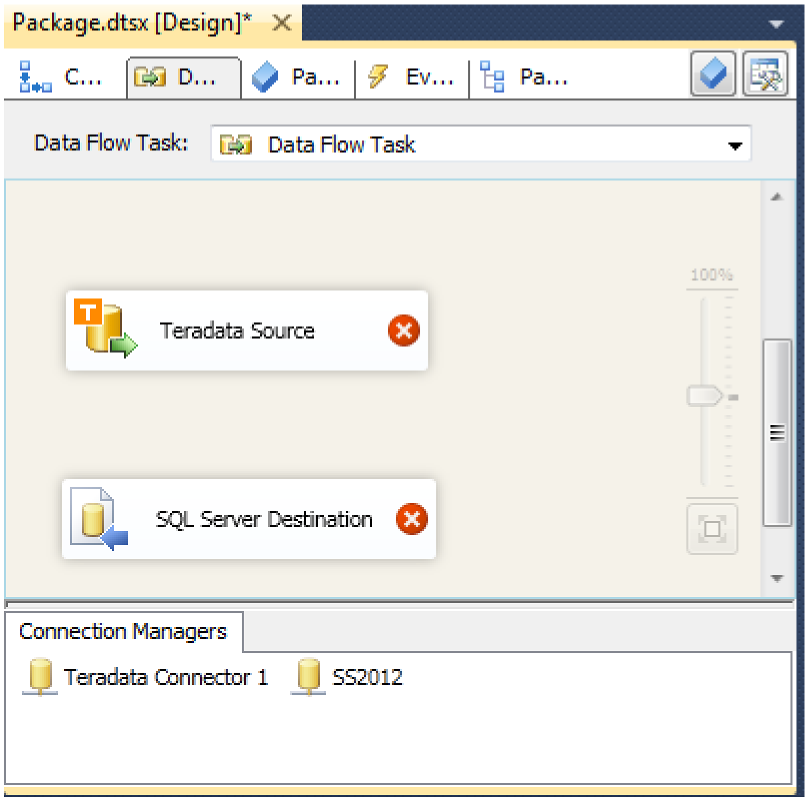


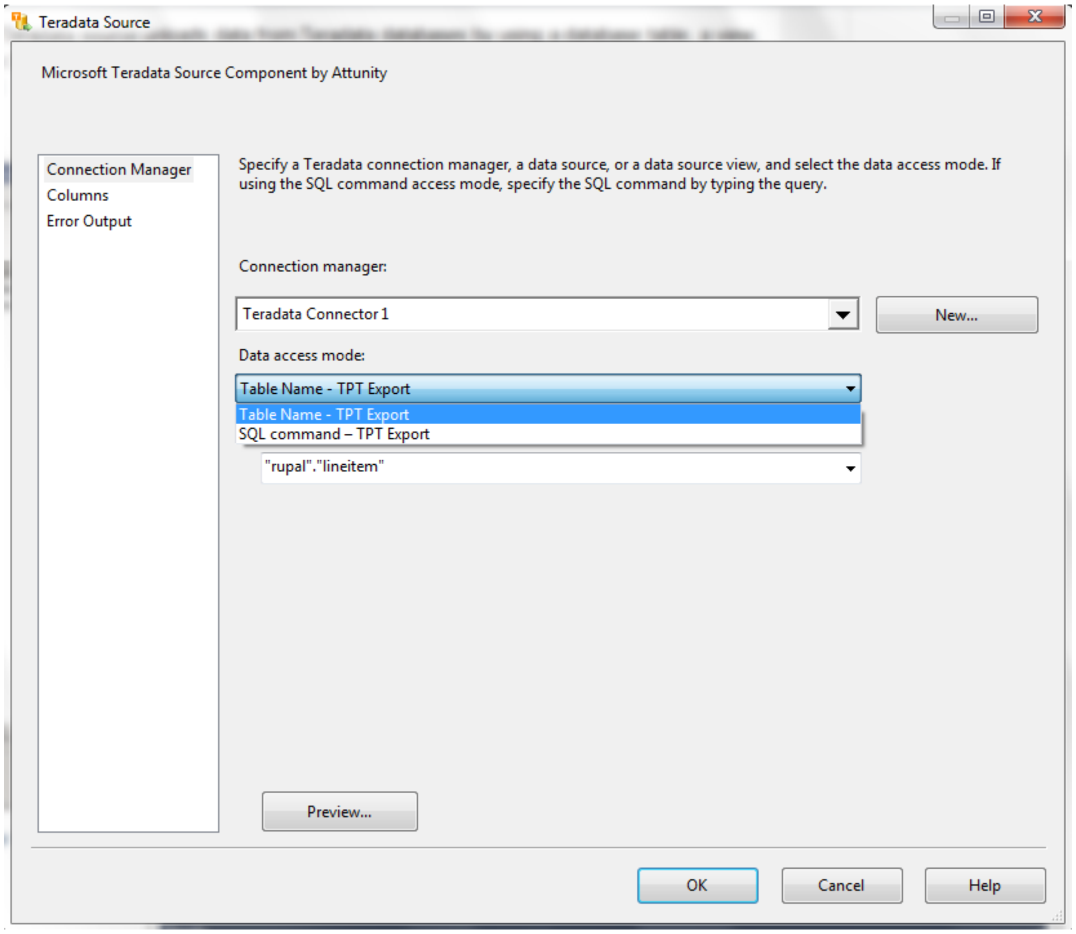
Figure 18: Source and destination in SSIS Designer

The source uses a Teradata connection manager, which specifies the Teradata provider to use. For more information, see Configuring the Teradata Connection Manager.

The Teradata source has one regular output and one error output.

The Teradata source unloads data from Teradata databases by using a database table, a view, or an SQL command. The Teradata source has the following data access modes for unloading data:

- A table or view.


Teradata Source

Microsoft Teradata Source Component by Attunity

Specify a Teradata connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command by typing the query.

Connection manager: Teradata Connector 1

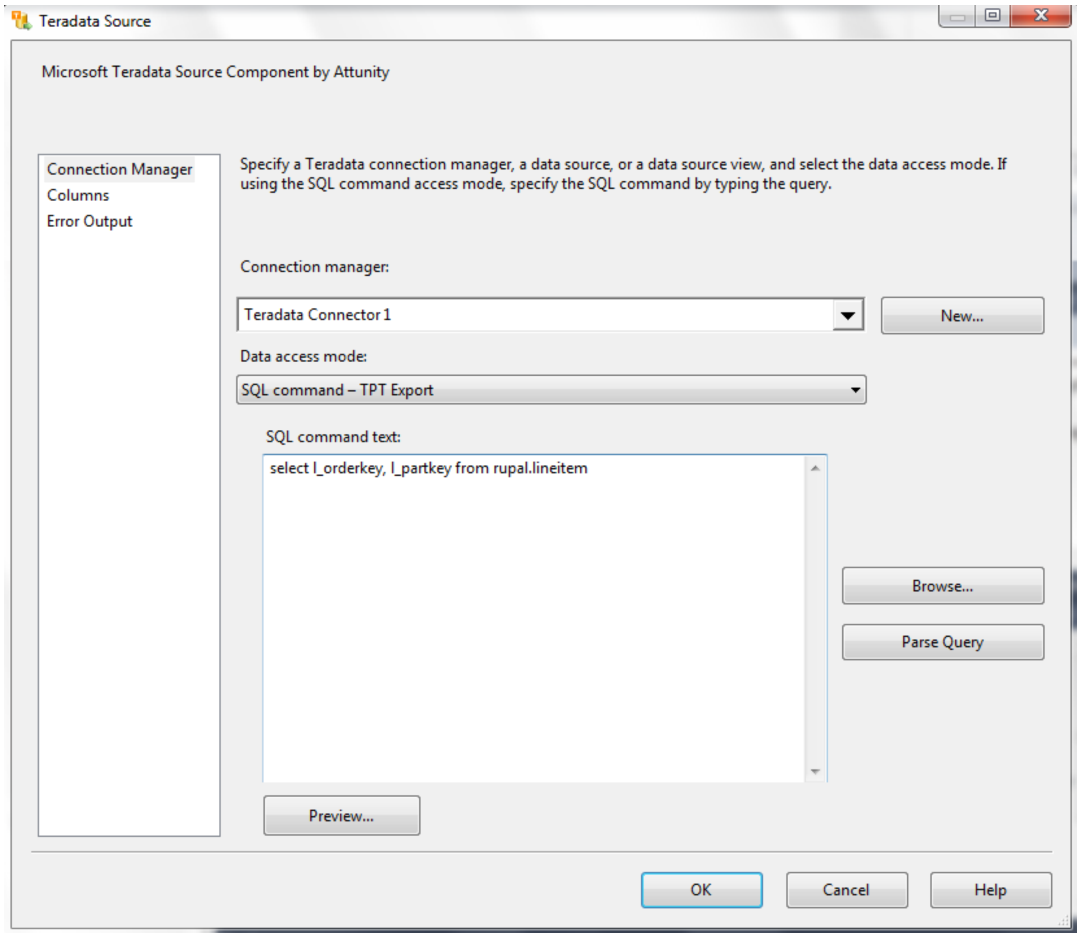
Data access mode: Table Name - TPT Export

Table Name - TPT Export
SQL command – TPT Export

"rupal"."lineitem"

Figure 19: Table Name access mode

• The results of an SQL statement.


Teradata Source

Microsoft Teradata Source Component by Attunity

Specify a Teradata connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command by typing the query.

Connection manager: Teradata Connector 1

Data access mode: SQL command – TPT Export

SQL command text:

```
select l_orderkey, l_partkey from rupal.lineitem
```

Figure 8: SQL Statement Access mode

Choosing the Data Access Mode

Select one of the following data access modes.

Option	Description
Table Name	Retrieve data from a table or view in the Teradata data source defined in the Connection Manager. If you select this option, select a value from the **Name of the table or the view** list. This list contains the first 1,000 tables only. If your database contains more than 1,000 tables, you can type the beginning of a table name or use the asterisk (*) wild card to enter any part of the name to display the table or tables you want to use.
SQL Statement	Retrieve data from the Teradata data source by using an SQL query. If you select this option, enter a query in one of the following ways: • Enter the text of the SQL query in the **SQL command** text field. • Click **Browse** to load the SQL query from a text file. Click **Parse query** to verify the syntax of the query text.

Adding and Configuring the Destination Component

After the Teradata Source component is set up, define the destination node. SQL Server Destination is a common one for this purpose. After the mapping is done, the SQL Server Integration Services package is ready to execute.

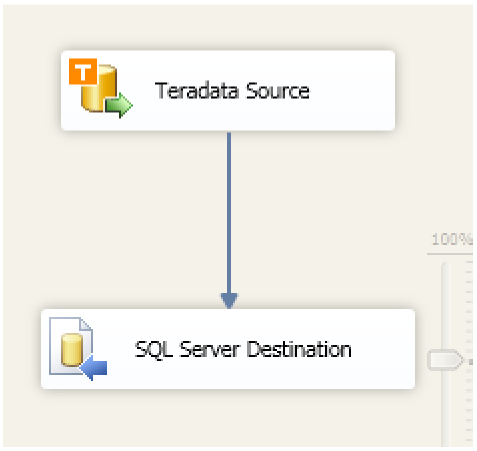


Figure 21: Data Flow task

Use Case 2: Bulk Load into Teradata Using TPT Load

This section describes a specific use case for the connector. It begins with an overview of the architecture and steps through configuration.

Use Case 2 Architecture Overview

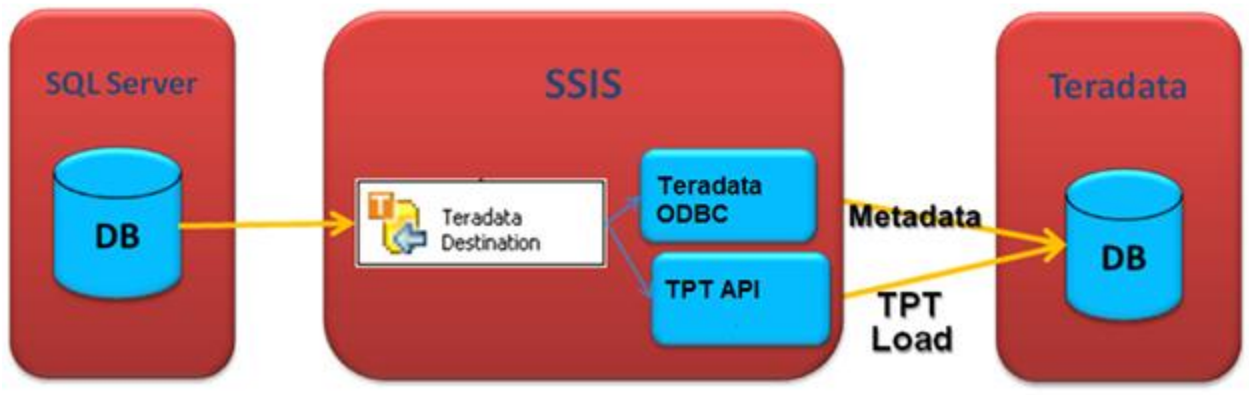


Figure 22: Use Case 2 architecture

Description

The challenge of loading large amounts of data can be solved by using the Microsoft Connector for Teradata by Attunity component in SQL Server Integration Services.

Defining the SQL Server Integration Services Data Flow

Adding and Configuring Source and Teradata Destination Components

The source can be any supported source. In figure 23, an OLE DB data source is selected.

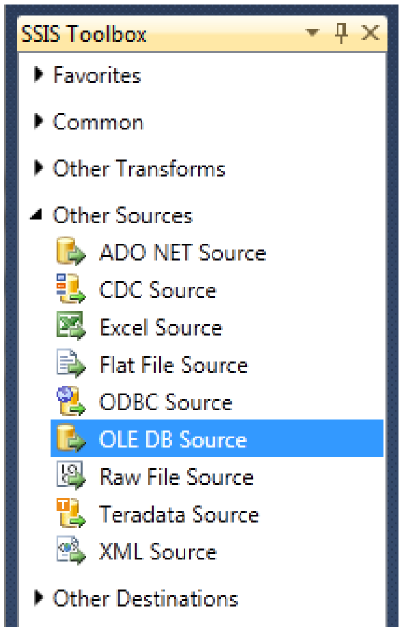


Figure 9: Choosing a data flow source

Add the Teradata Destination Component.

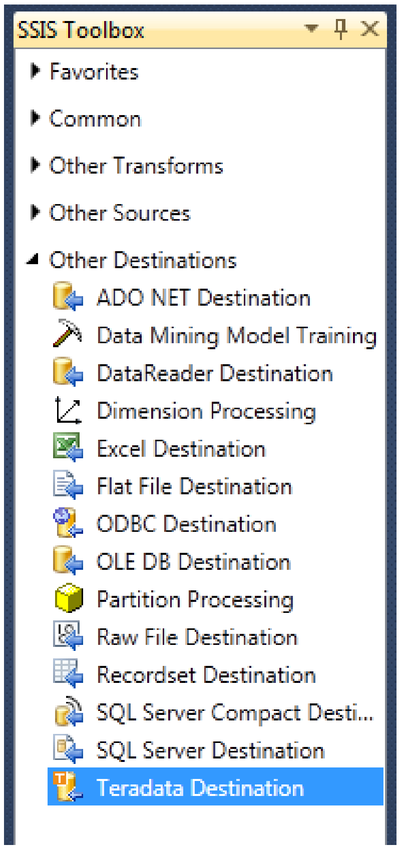


Figure 24: Adding a destination

Now, link the source and the destination.

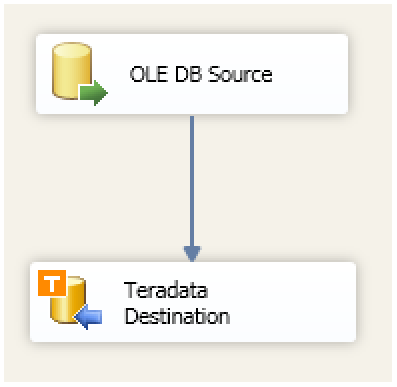
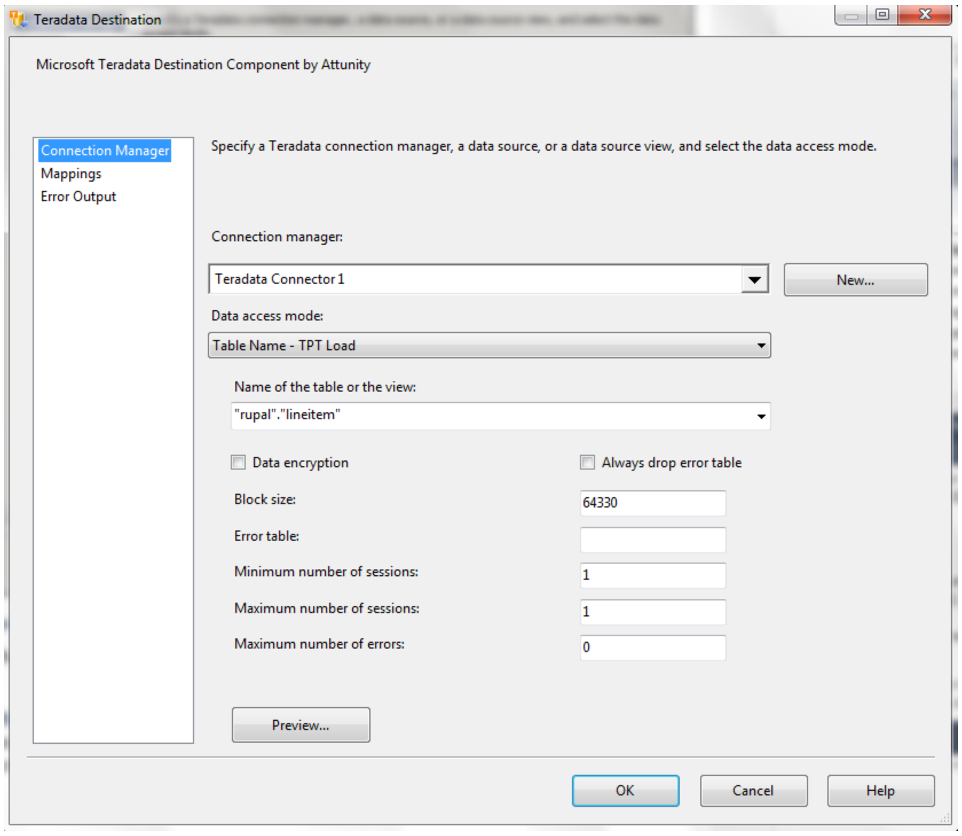


Figure 25: Linking source and destination

Configure the Teradata Destination Component. For more information, see Appendix C – Teradata Destination Component Advanced Topics.


Teradata Destination

Microsoft Teradata Destination Component by Attunity

Specify a Teradata connection manager, a data source, or a data source view, and select the data access mode.

Connection manager: Teradata Connector 1

Data access mode: Table Name - TPT Load

Name of the table or the view: "rupal"."lineitem"

Data encryption [ ]

Always drop error table [ ]

Block size: 64330

Error table:

Minimum number of sessions: 1

Maximum number of sessions: 1

Maximum number of errors: 0

Figure 26: Teradata Destination Component

After the mapping is done, the SQL Server Integration Services package is ready to execute.

Use Case 3: Incremental Load into Teradata Using TPT Stream

This section describes a specific use case for the connector. It begins with an overview of the architecture and steps through configuration.

Description

The Teradata destination connects to a Teradata database and incrementally (batch) loads data into Teradata databases using the TPT Stream operator. This mode enables you to load to an existing Teradata table with data.

Note, this release of the connector does not allow TPT Stream to submit updates, deletes or upserts to empty or existing Teradata Database tables.

The destination uses the Teradata connection manager to connect to a data source.

Architecture

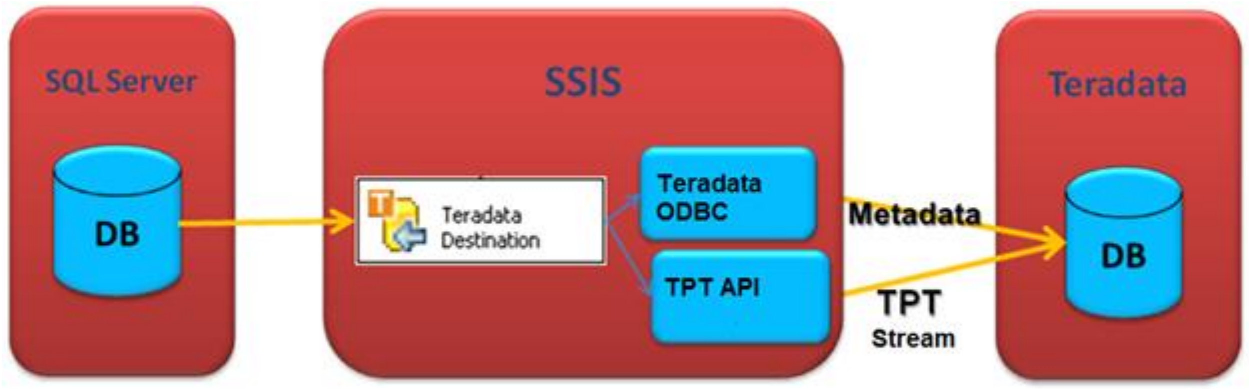


Figure 27: Use Case 3 architecture

Defining the SQL Server Integration Services Data Flow

Adding and Configuring Source and Teradata Destination Components

The source can be any supported source. In figure 28, an OLE DB data source is selected.

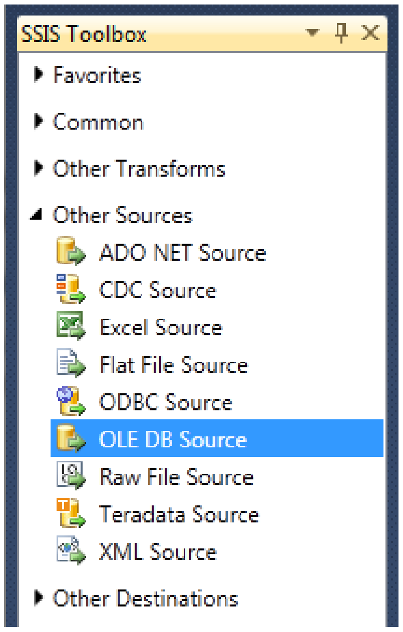


Figure 28: Choosing a data flow source

Add the Teradata Destination Component.

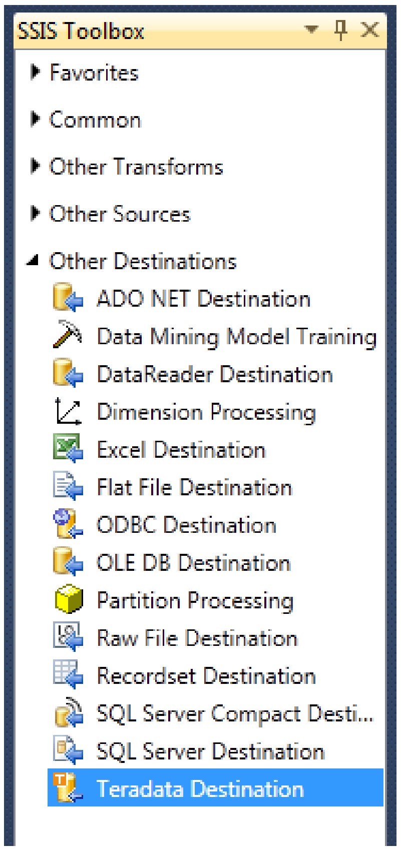


Figure 29: Adding a destination

Now, link the source and the destination.

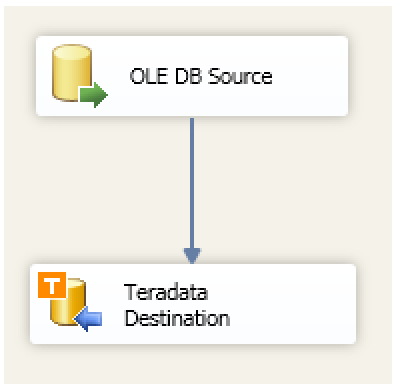
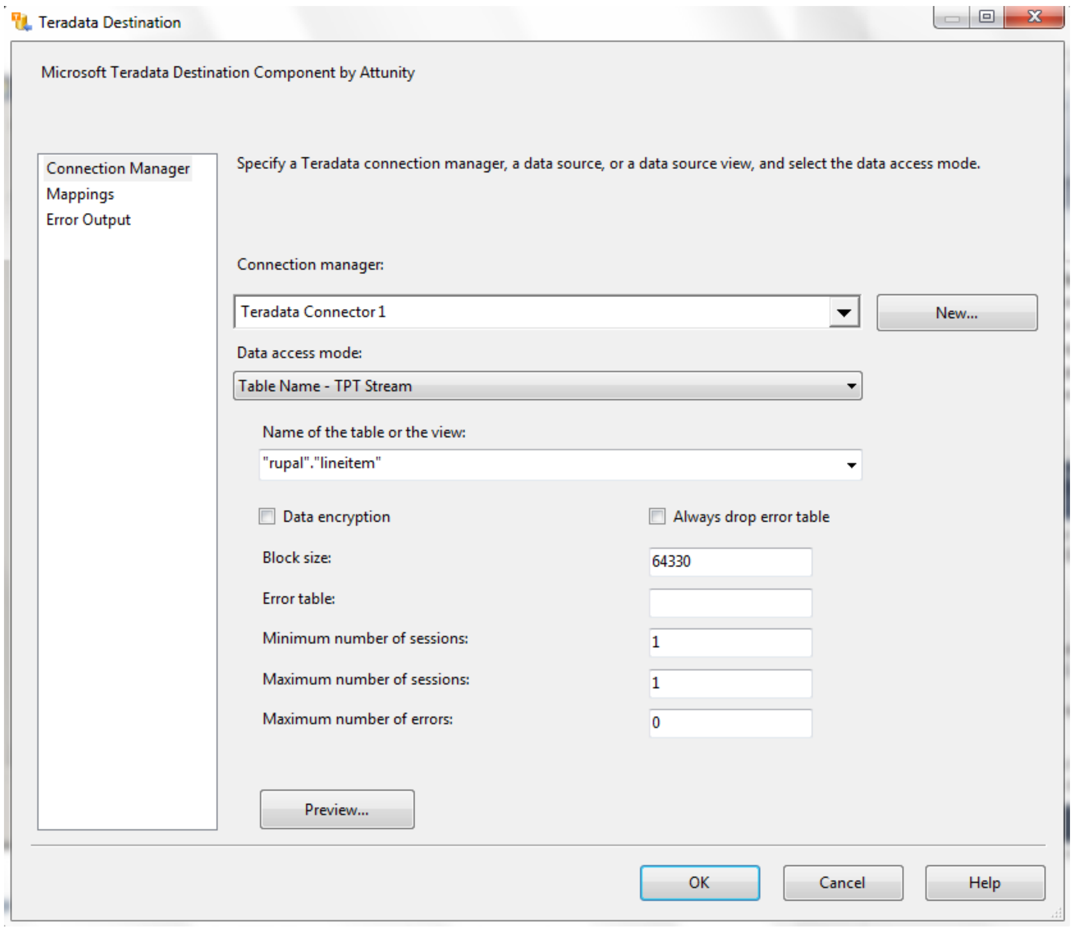


Figure 30: Linking source and destination

Configure the Teradata Destination Component. For more information, see [Appendix C – Teradata Destination Component Advanced Topics]().


Teradata Destination

Microsoft Teradata Destination Component by Attunity

Specify a Teradata connection manager, a data source, or a data source view, and select the data access mode.

Connection manager: Teradata Connector 1

Data access mode: Table Name - TPT Stream

Name of the table or the view: "rupal"."lineitem"

☐ Data encryption

☐ Always drop error table

Block size: 64330

Error table:

Minimum number of sessions: 1

Maximum number of sessions: 1

Maximum number of errors: 0

Figure 31: Teradata Destination Component – TPT Stream

After the mapping is done, the SQL Server Integration Services package is ready to execute.

Conclusion

The Microsoft Connector for Teradata by Attunity provides a high-performance means of loading and unloading data to and from Teradata databases. This paper discusses the functionality of the connector, and it provides detailed step-by-step instructions on how to use the connector with SQL Server Integrated Services. Three general use cases are presented with the design highlights.

Additional detailed technical information is contained in the appendices that follow.

For more information:

SQL Server Integration Services Web site: http://www.microsoft.com/en-us/sqlserver/solutions-technologies/enterprise-information-management/integration-services.aspx

SQL Server Integration Services TechCenter: http://technet.microsoft.com/en-us/library/ms141026.aspx

SQL Server Integration Services DevCenter: http://msdn.microsoft.com/en-us/library/ms141026.aspx

Attunity User Forums-Teradata Connector: http://www.attunity.com/forums/micorosft-ssis-teradata-connector

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

- Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
- Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

Send feedback.

Appendix A – Data Types

Supported Data Types

The SQL Server Integration Services components for Teradata use the Teradata Parallel Transporter API library with the Load, Stream, and Export operators to fully utilize the Teradata parallel processing capabilities. Because the SQL Server Integration Services components for Teradata use the Teradata Parallel Transporter API to load and unload data from Teradata databases, only data types supported by the API can be used with these components.

Columns of unsupported data types are shown but cannot be mapped. Tables with columns of unsupported data types that are not nullable cannot be loaded.

Note: Teradata has many TIME, TIMESTAMP, and INTERVAL data types. These data types are handled by Teradata Parallel Transporter as fixed-sized character strings. They are handled by the SQL Server Integration Services components for Teradata as strings.

Data Type Mapping

The following table shows the Teradata database data types and their default mapping to SQL Server Integration Services data types. It also shows the unsupported data types.

Teradata data type	SQL Server Integration Services data type
Decimal/backend data types	
DECIMAL/NUMERIC	DT_NUMERIC
BYTEINT	DT_I1
SMALLINT	DT_I2
INTEGER	DT_I4
FLOAT/REAL/DOUBLE PRECISION	DT_R8
Date/time and interval data types	
DATE	DT_DBDATE
TIME	DT_STR

TIMESTAMP TIMESTAMP (n)	DT_STR
TIME WITH TIMEZONE TIME(n) WITH TIMEZONE	DT_STR
TIMESTAMP WITH TIMEZONE TIMESTAMP(n) WITH TIMEZONE	DT_STR
INTERVAL YEAR INTERVAL YEAR (n)	DT_STR
INTERVAL YEAR TO MONTH INTERVAL YEAR (n) TO MONTH	DT_STR
INTERVAL MONTH INTERVAL MONTH (n)	DT_STR
INTERVAL DAY INTERVAL DAY (n)	DT_STR
INTERVAL DAY TO HOUR INTERVAL DAY (n) TO HOUR	DT_STR
INTERVAL DAY TO MINUTE INTERVAL DAY (n) TO MINUTE	DT_STR
INTERVAL DAY TO SECOND INTERVAL DAY (n) TO SECOND INTERVAL DAY TO SECOND (m) INTERVAL DAY (n) TO SECOND (m)	DT_STR

INTERVAL HOUR INTERVAL HOUR (n)	DT_STR
INTERVAL HOUR TO MINUTE INTERVAL HOUR (n) TO MINUTE	DT_STR
INTERVAL HOUR TO SECOND INTERVAL HOUR (n) TO SECOND INTERVAL HOUR TO SECOND (m) INTERVAL HOUR (n) TO SECOND (m)	DT_STR
INTERVAL MINUTE INTERVAL MINUTE (n)	DT_STR
INTERVAL MINUTE TO SECOND INTERVAL MINUTE (n) TO SECOND INTERVAL MINUTE TO SECOND (m) INTERVAL MINUTE (n) TO SECOND (m)	DT_STR
INTERVAL SECOND INTERVAL SECOND (n) INTERVAL SECOND (n,m)	DT_STR
Character data types	
CHARACTER	DT_STR
VARCHAR	DT_STR
LONG VARCHAR	DT_STR **Note:** The data is truncated to the maximum allowed size for DT_STR, which is 8,000

	characters.
CLOB	Not supported
Byte data types	
BYTE	DT_BYTES
VARBYTE	DT_BYTES
BLOB	Not supported

Appendix B – Teradata Source Advanced Topics

Teradata Source

The Teradata source uses the Export operator to unload data from Teradata databases by using a database table, a view, or an SQL command. The Teradata source has the following data access modes for unloading data:

- A table or view
- The results of an SQL statement

The source uses a Teradata connection manager, which specifies the Teradata provider to use.

The Teradata source has one regular output and one error output.

Error Handling

The Teradata source has an error output. The component error output includes the following output columns:

- **Teradata Error Code**: The number that corresponds to the current error. For more information, including a list of relevant error codes, see the Teradata documentation.
- **Error Column**: The source column causing the error (for conversion errors).
- **Error Row Columns**: The record data that causes the error.

Depending on the error behavior setting, the Teradata source supports returning errors (data conversion, truncation) that occur during the unloading process in the error output.

Working with Teradata Stream and Load protocols, the errors that occur during the load process are written by Teradata to temporary error tables that are locked during the loading process.

Parallelism

Multiple independent Export jobs can access the same table or different tables at the same time. The Teradata database limits the number of Export jobs that can run at the same time. This limit is set in a database variable called *MaxLoadTasks*. Therefore, you can execute more than one Teradata source at the same time to unload data from one or more tables. You can define the maximum number of Teradata sources that can run in parallel with the **MaxLoadTasks** property. This property defines the maximum number of Teradata sources that can run at the same time.

Troubleshooting the Teradata Source

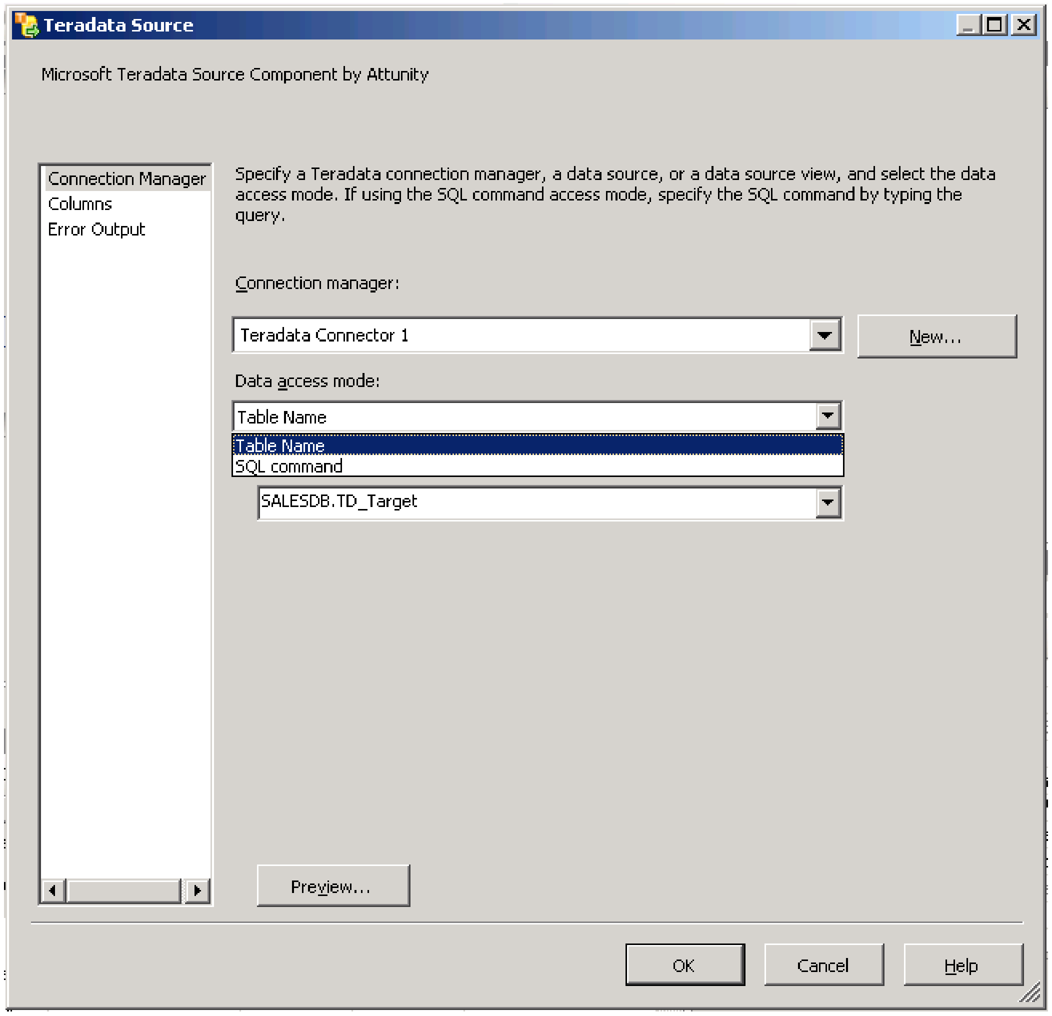
You can log the calls that the Teradata source makes to the Teradata Parallel Transporter API library. You can use this logging ability to troubleshoot the unloading of data from Teradata data sources that the Teradata source performs. To log the calls that the Teradata source makes to a Teradata data source, enable package logging and select the Diagnostic event at the package level.

Configuring the Teradata Source

You can configure the Teradata Source programmatically or through the SSIS Designer. The following sections contain information about how to do this.

Teradata Source Editor (Connection Manager Page)

Use the **Connection Manager** page of the Teradata Source Editor to select the Teradata connection manager for the source. This page also lets you select a table or view from the database.



Teradata Source

Microsoft Teradata Source Component by Attunity

Specify a Teradata connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command by typing the query.

Connection manager: Teradata Connector 1

Data access mode: Table Name

SALESDB.TD_Target

Figure 32: Teradata Source – Connection Manager

Options

Connection Manager

Select an existing connection manager from the list, or click **New** to create a new connection. The Teradata Connection Manager Editor opens, where you can create a new connection manager.

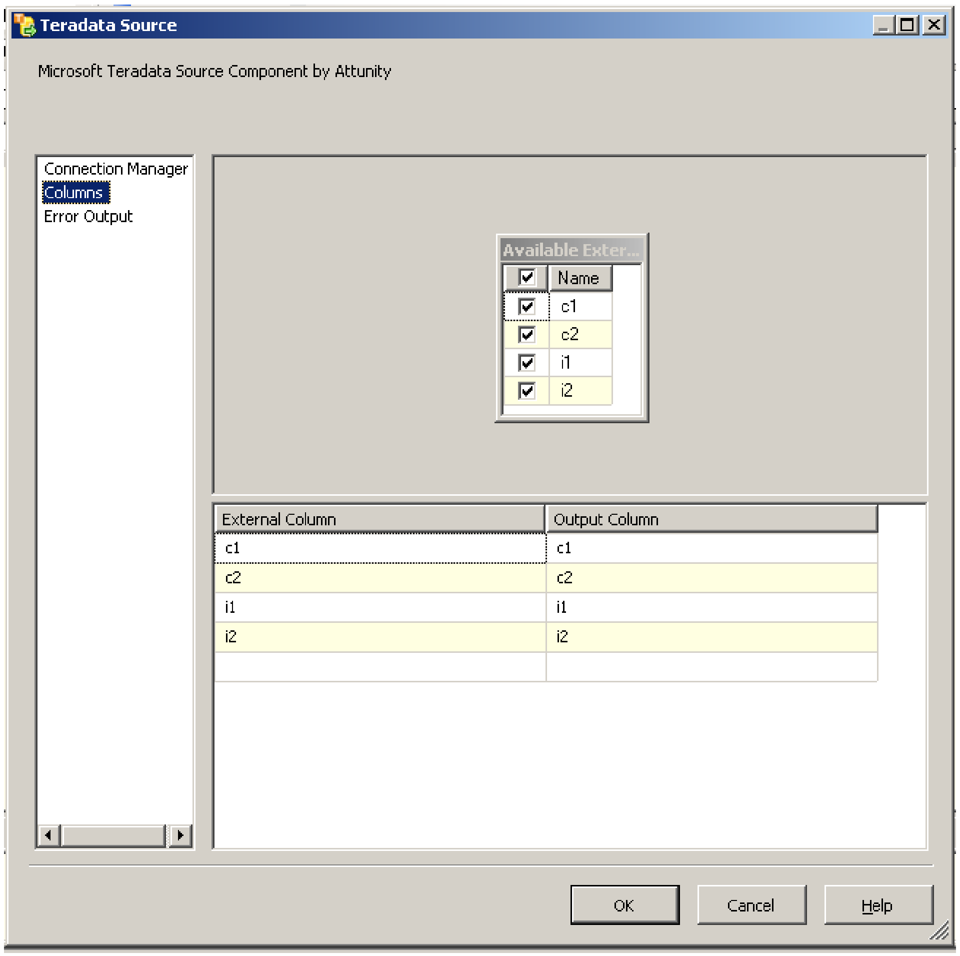
Data Access Mode

Select the method for selecting data from the source. The options are shown in the following table.

Option	Description
Table or view	Retrieve data from a table or view in the Teradata data source. When you select this option, select a value from the drop-down list for the name of the table or the view. Select an available table or view from the database from the list. This list contains the first 1,000 tables only. If your database contains more than 1,000 tables, you can type the beginning of a table name or use the (*) wild card to enter any part of the name to display the table or tables you want to use.
SQL command	Retrieve data from the Teradata data source by using an SQL query. When you select this option, enter a query in one of the following ways: • Enter the text of the SQL query in the **SQL command text** field. • Click **Browse** to load the SQL query from a text file. To verify the syntax of the query text, click **Parse query**.

Teradata Source Editor (Columns Page)

Use the **Columns** page of the Teradata Source Editor to map an output column to each external (source) column.



✓	Name
✓	c1
✓	c2
✓	i1
✓	i2

External Column	Output Column
c1	c1
c2	c2
i1	i1
i2	i2

Figure 33: Teradata Source Component - COLUMNS

Options

Available External Columns

You cannot use this interface to add or delete columns. Select the columns to use in the source. The selected columns are added to the **External Column** list in the order in which you select them.

Select the **Select All** check box to select all of the columns.

External Column

To change the order of columns, first clear the selected columns in the **Available External Columns** list, and then select external columns from the list in a different order. The selected columns are added to the **External Column** list in the order in which you select them.

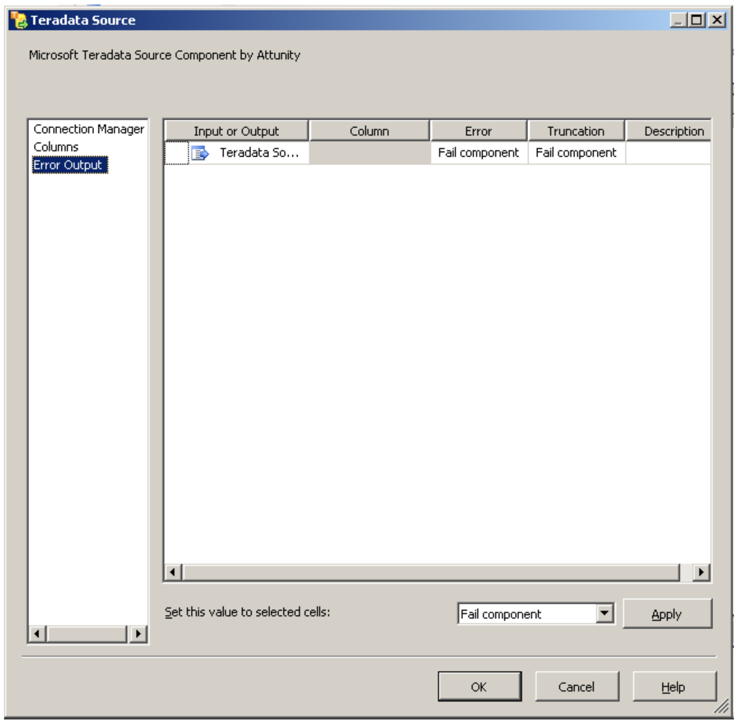
Output Column

Enter a unique name for each output column. The default is the name of the selected external (source) column; however, you can choose any unique, descriptive name. The name entered is displayed in Business Intelligence Development Studio.

Note: Columns of unsupported data types are shown as external columns, but they are not exposed as output columns.

Teradata Source Editor (Error Output Page)

Use the **Error Output** page of the Teradata Source Editor to select error-handling options.


Teradata Source
Microsoft Teradata Source Component by Attunity
Connection Manager
Columns
Error Output
Set this value to selected cells:
Apply
OK
Cancel
Help

Input or Output	Column	Error	Truncation	Description
Teradata So...		Fail component	Fail component	

Set this value to selected cells: Fail component

Figure 34: Teradata Source Component - Error output

Options

Error Behavior

Select how the Teradata source should handle errors in a flow: ignore the failure, redirect the row, or fail the component.

Truncation

Select how the Teradata source should handle truncation in a flow: ignore the failure, redirect the row, or fail the component.

Error-Handling Options

You use the following options to configure how the Teradata source handles errors and truncations.

Fail Component

The Data Flow task fails if an error or a truncation occurs. This is the default behavior.

Ignore Failure

The error or the truncation is ignored and the data row is directed to the Teradata source output.

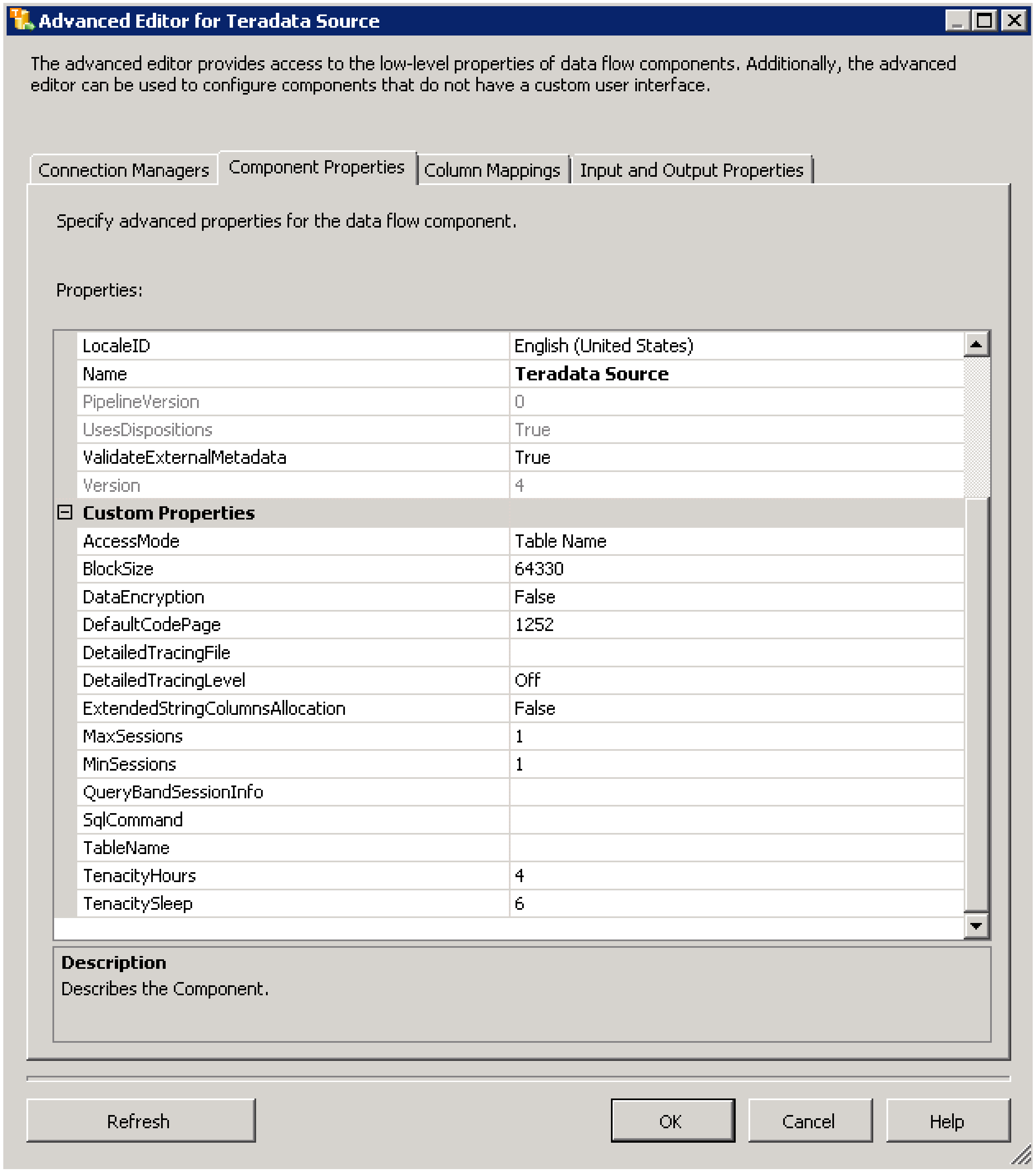
Redirect Flow

The error or the truncation data row is directed to the error output of the Teradata source. In this case the Teradata source error handling is used.

Teradata Source Advanced Editor

The Advanced Editor contains properties that can be set programmatically. To open the Advanced Editor:

- In the **Data Flow** screen of your SQL Server Integration Services project, right-click the Teradata source and then click **Show Advanced Editor**.


Advanced Editor for Teradata Source

The advanced editor provides access to the low-level properties of data flow components. Additionally, the advanced editor can be used to configure components that do not have a custom user interface.

Connection Managers | Component Properties | Column Mappings | Input and Output Properties

Specify advanced properties for the data flow component.

Properties:

LocaleID	English (United States)
Name	**Teradata Source**
PipelineVersion	0
UsesDispositions	True
ValidateExternalMetadata	True
Version	4
Custom Properties	
AccessMode	Table Name
BlockSize	64330
DataEncryption	False
DefaultCodePage	1252
DetailedTracingFile	
DetailedTracingLevel	Off
ExtendedStringColumnsAllocation	False
MaxSessions	1
MinSessions	1
QueryBandSessionInfo	
SqlCommand	
TableName	
TenacityHours	4
TenacitySleep	6

Description
Describes the Component.

Figure 35: Teradata Source Advanced Component properties

Teradata Source Custom Properties

The following table describes the custom properties of the Teradata source. All properties are read/write.

Property name	**Data type**	**Description**
AccessMode	Integer (Enumeration)	The mode used to access the database. The possible options are **TableName** and **SqlCommand**. The default is **TableName**.
BlockSize	Integer	The block size, in bytes, used when returning data to the client. The minimum value is 256 bytes. The default and maximum value is 64,330 bytes. **Note**: This property is available in the Advanced Editor.
DataEncryption	Boolean	Indicates whether full security encryption of SQL requests, responses, and data is used: • If this property is not selected, no encryption is used. **This is the default setting.** • If this property is selected, all SQL requests, responses, and data are encrypted. **Note**: This property is available in the Advanced Editor.
DefaultCodePage	Integer	The code page to use when code page information is unavailable from the data source. **Note**: This property is available in the Advanced Editor.
DetailedTracing File	String	The path that indicates the physical location of the log file. A log file is generated automatically when any **DetailedTracingLevel** value (except **Off**) is selected. **Note**: This property is available in the Advanced Editor.
DetailTracingLevel	Integer	This allows setting the TPT API tracing to different levels. The default value is **Off** **Note**: This property is available in the Advanced Editor.

Property name	Data type	Description
ExtendedStringColumnsAllocation	Boolean	A value that indicates whether the Maximal Transfer Character Allocation Factor is used. This value should be set to **True** if the Teradata database **Export Width Table ID** property is set to **Maximal Defaults**. The default value is **False**. **Note**: This property is available in the Advanced Editor.
MaxSessions	Integer	The maximum number of sessions that are logged on. This value must be greater than zero. The default value is one session for each available Access Module Processor (AMP). **Note**: This property is available in the Advanced Editor.
MinSessions	Integer	The minimum number of sessions that are logged on. This value must be greater than zero. The default value is one session for each available AMP. **Note**: This property is available in the Advanced Editor.
QueryBandSessionInfo	String	Enables a user-defined query band expression that is set for every SQL session connected by the Teradata PT operator. This allows charge back, monitoring, and governance. **Note**: This property is available in the Advanced Editor.
SqlCommand	String	The SQL command to be executed.
TableName	String	The name of the table with the data that is being used.

Property name	Data type	Description
TenacityHours	Integer	The number of hours the driver attempts to log on when the maximum number offload/export operations is already running. The default is 4 hours. **Note**: This property is available in the Advanced Editor.
TenacitySleep	Integer	Specifies the number of minutes the driver pauses before attempting to log on under the restraints defined by the **MaxSessions** and **TenacityHours** properties. The default is 6 minutes. **Note**: This property is available in the Advanced Editor.

Appendix C – Teradata Destination Component Advanced Topics

Teradata Destination

The Teradata destination connects to a local or remote Teradata database and bulk loads data into Teradata databases.

The destination uses a Teradata connection manager to connect to a data source. For more information, see Configuring the Teradata Connection Manager.

A Teradata destination includes mappings between input columns and columns in the destination data source. You do not have to map input columns to all destination columns, but depending on the properties of the destination columns, errors can occur if no input columns are mapped to the destination columns. For example, if a destination column does not allow null values, an input column must be mapped to that column. In addition, the data types of mapped columns must be compatible. For example, you cannot map an input column with a string data type to a destination column with a numeric data type.

The Teradata destination has one regular input and one error output.

Note: Columns of unsupported data types are shown, but cannot be mapped. Tables that have columns of unsupported data types that are not nullable cannot be loaded. For more information, see Supported Data Types.

Load Options

The Teradata destination can use one of two access load modes. You set the mode in the Teradata Destination Editor (Connection Manager Page). The two modes are:

- Incremental loading: This mode uses the Teradata Stream operator. This mode is used if **Access Mode** is set to **Table Name – TPT Stream**. For more information about how to set the properties for this mode, see Teradata Destination Editor (Connection Manager Page) and Teradata Destination Custom Properties.
- For bulk loading, use TPT Load: In this mode, the destination component uses the TPT Load protocol for fast bulk loading the Teradata table.

Note: If you want to use TPT Load, the destination Teradata table must be empty.

Error Handling

The Teradata Stream and Load operators write errors that occur during the load process to temporary error tables that are locked during the loading process. You can set the maximum number of errors that can be written to these tables in the **Maximum number of errors** property. This property is defined programmatically in the Advanced Editor. For more information, see Teradata Destination Custom Properties.

When you are executing the Teradata destination, if the value of **Maximum number of errors** is greater than zero, unique names are generated for the error tables, an informational message with the generated names to the package log is printed, and the bulk loading process begins.

When this process is complete, the Teradata destination uses the ODBC driver for Teradata to access the tables and retrieve the error information. The errors are returned in a SQL Server Integration Services component error output, depending on the error behavior setting. After completing the process, the component drops the temporary tables.

If the Teradata destination is stopped before it completes the process, the temporary tables are not dropped. In this case, you can use an SQL task to drop the tables manually, if necessary.

The Teradata destination generates one of two types of error tables. The type of table depends on the load options mode being used.

The component error output includes the following output columns, if incremental loading (Stream operator) is used:

- **Teradata Error Code**: The number that corresponds to the current error. See the Teradata documentation for a list of relevant error codes.
- **Error Message**: The message that accompanies the error code and describes the error.
- **Error Column**: The source column that caused the error (for conversion errors).
- **Source Sequence**: The sequence number of the source line that caused the error.

The component error output includes the following output columns if the fast load (fast bulk load) protocol (TPT Load operator) is used:

- **Teradata Error Code**: The number that corresponds to the current error. See the Teradata documentation for a list of relevant error codes.
- **Error Message**: The message that accompanies the error code and describes the error.
- **Error Column**: The source column that caused the error (for conversion errors).
- **Error Row Columns**: The record data that caused the error.

Depending on the error behavior setting, the Teradata destination supports returning errors (data conversion, truncation, or constraint violation) that occur during the unloading process in the error output.

When the destination component returns the number of errors that is set in the **Maximum number of errors** property, the last error is returned and the data flow task fails. The target table state depends on the mode being used. If you are using the Load operator, the target table is not usable because the job did not finish correctly. The Teradata destination drops the error tables and you must truncate or drop and then re-create the target table, fix the errors, and then execute the Teradata destination again. Rollback is not supported when the load mode is used, because the target table must be empty.

If you are using incremental loading mode (Stream operator), there is no rollback concept. When the Teradata destination executes, rows are buffered. When the buffer is full, it is sent to the database and at that time, all of the changes made by those rows are committed. If the job fails in this mode, all of the changes that were completed at that time of the failure (depending on when the buffers were sent) are present in the target table(s) and are not rolled back. The Teradata destination will drop the error tables.

Parallelism

When the Load operator is used, it locks the destination table. Hence, you cannot run multiple load jobs against the same table. Parallelism on the client side is restricted in the current version of the Attunity connector until the Instance attribute is implemented. Therefore, running more than one Teradata destination at the same time with the Load operator enabled can be done only if different tables are loaded. The Teradata database also limits the number of load jobs that can run at the same time. This limit is set in a database variable called *MaxLoadTasks*.

When incremental loading is used (Stream operator), there is no restriction on the number of Teradata destinations that run in parallel against the same tables. Incremental-loading Teradata destinations do not count in the **MaxLoadTasks** limit. Therefore, it is possible to run multiple Teradata destinations with fast load disabled on the same database table. The number of components that can run concurrently is not restricted by general database session parameters.

Note: Although it is possible to run multiple Teradata destinations concurrently against the same table when working in incremental loading mode (Stream operator), this does not mean that this is an effective way to work. According to Teradata, doing this can actually reduce performance (the Stream parallelism occurs on the server side; therefore, running multiple Stream jobs on the same table may lead to lock contentions).

Troubleshooting the Teradata Destination

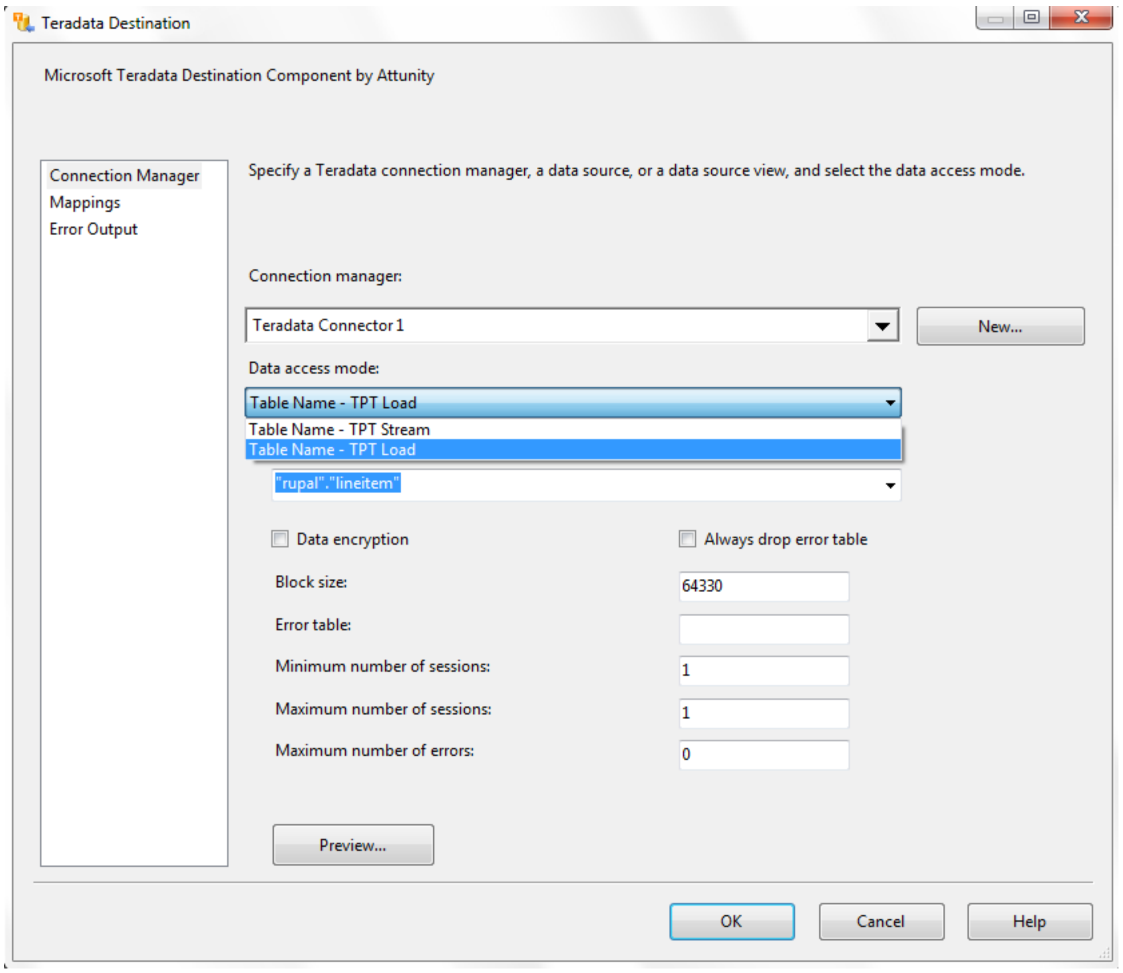
You can log the calls that the Teradata destination makes to the Teradata Parallel Transporter API library. You can use this logging ability to troubleshoot the saving of data to Teradata data sources that the Teradata destination performs. To log the calls that the Teradata destination makes to a Teradata data source, enable package logging and select the Diagnostic event at the package level.

Configuring the Teradata Destination

You can configure the Teradata destination programmatically or through SSIS Designer.

Teradata Destination Editor (Connection Manager Page)

Use the **Connection Manager** page of the Teradata Destination Editor to select the Teradata connection manager for the destination. You can also select a table or view from the database.


Teradata Destination

Microsoft Teradata Destination Component by Attunity

Specify a Teradata connection manager, a data source, or a data source view, and select the data access mode.

Connection manager: Teradata Connector 1

Data access mode: Table Name - TPT Load

- Table Name - TPT Stream
- Table Name - TPT Load

"rupal"."lineitem"

Data encryption: ☐

Always drop error table: ☐

Block size: 64330

Error table:

Minimum number of sessions: 1

Maximum number of sessions: 1

Maximum number of errors: 0

Figure 36: Teradata Destination Component – Connection Manager

Options

Connection manager

Select an existing connection manager from the list, or click **New** to create a new connection. The Teradata Connection Manager Editor opens where you can create a new connection manager.

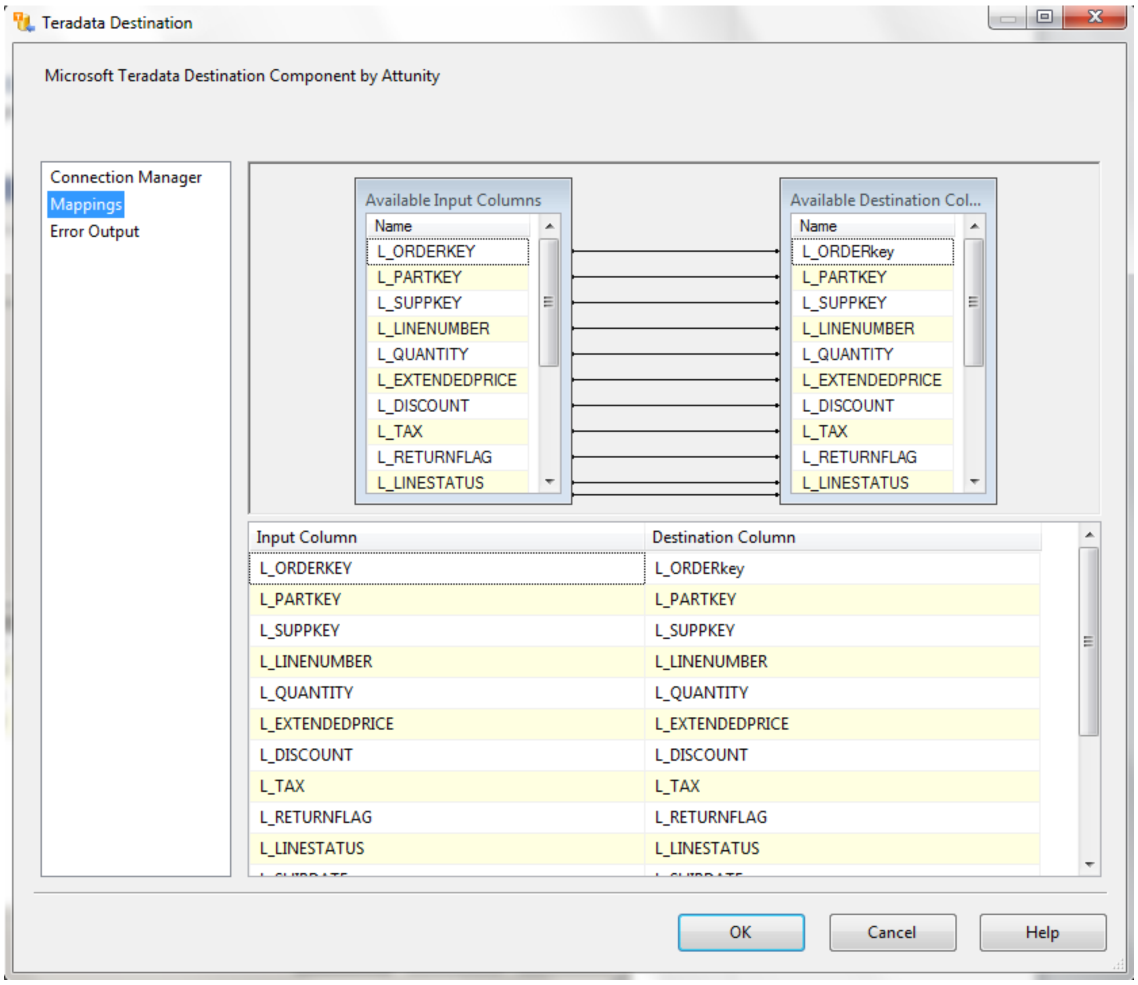
Data access mode

Select the method for selecting data from the source. The options are shown in the following table.

Option	Description
Table Name – TPT Stream	Select this option to configure the Teradata destination to work in arrayed mode. When you select this option, the following options are available: • **Data Encryption**: Indicates whether full security encryption of SQL requests, responses, and data is used: o If this property is not selected, no encryption is used. **This is the default value.** o If this property is selected, all SQL requests, responses, and data are encrypted. • **Always Drop Error Table**: A value that indicates whether all error tables are dropped even if the Teradata destination fails to read the data. The default value is **False**. • **Block size**: The block size, in bytes, used when data is returned to the client. The minimum value is 256 bytes. The default and maximum value is 64,330 bytes. • **Error Table**: The name indicator used to create names for the generated error tables. The default value is the target table name. • **Minimum number of sessions**: The minimum number of sessions that are logged on. This value must be greater than zero. The default value is one session for each available AMP. • **Maximum number of sessions**: The maximum number of sessions that are logged on. This value must be greater than zero. The default value is one session for each available AMP. • **Maximum number of errors**: The number of errors that can occur before the data flow is stopped. By default the value is 0, which indicates that there is no error limit. All errors that are returned before the data flow reaches the error limit are returned.
Table Name-TPT Load	Select this option to configure the Teradata destination to work in FASTLOAD load mode. The same options as in the TPT Stream are available for this mode.

Teradata Destination Editor (Mappings Page)

Use the **Mappings** page of the Teradata Destination Editor to map input columns to destination columns.



Name
L_ORDERKEY
L_PARTKEY
L_SUPPKEY
L_LINENUMBER
L_QUANTITY
L_EXTENDEDPRICE
L_DISCOUNT
L_TAX
L_RETURNFLAG
L_LINESTATUS

Name
L_ORDERkey
L_PARTKEY
L_SUPPKEY
L_LINENUMBER
L_QUANTITY
L_EXTENDEDPRICE
L_DISCOUNT
L_TAX
L_RETURNFLAG
L_LINESTATUS

Input Column	Destination Column
L_ORDERKEY	L_ORDERkey
L_PARTKEY	L_PARTKEY
L_SUPPKEY	L_SUPPKEY
L_LINENUMBER	L_LINENUMBER
L_QUANTITY	L_QUANTITY
L_EXTENDEDPRICE	L_EXTENDEDPRICE
L_DISCOUNT	L_DISCOUNT
L_TAX	L_TAX
L_RETURNFLAG	L_RETURNFLAG
L_LINESTATUS	L_LINESTATUS

Figure 37: Teradata Destination Component - Mappings

Options

Available Input Columns

The list of available input columns. Drag an input column to an available destination column to map the columns.

Note: Columns of unsupported data types are shown, but they cannot be mapped.

Available Destination Columns

Drag a destination column to an available input column to map the columns.

Note: Columns of unsupported data types are shown, but they cannot be mapped.

Input Column

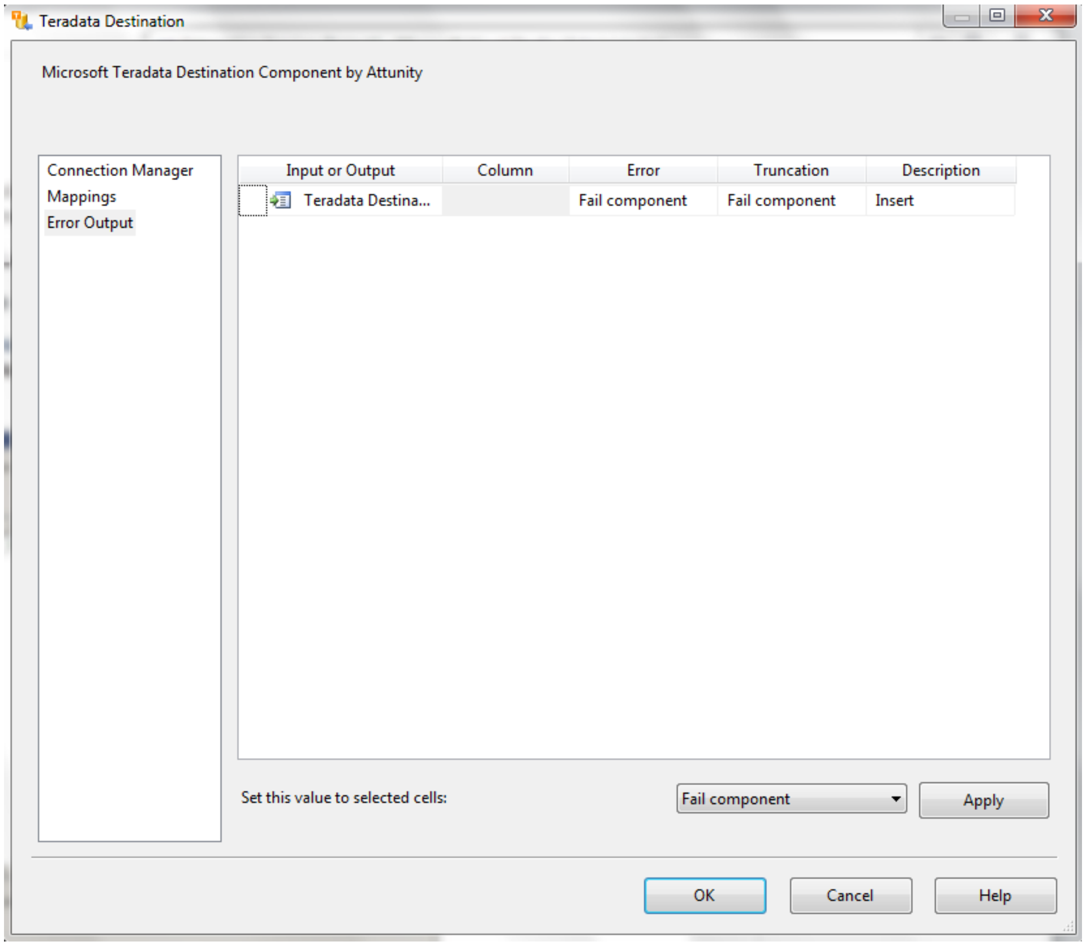
View the input columns that you selected. To remove mappings by excluding columns from the output, click **<ignore>**.

Destination Column

View all available destination columns, both mapped and unmapped.

Teradata Destination Editor (Error Output Page)

Use the **Error Output** page of the Teradata Destination Editor to configure error-handling options.



Input or Output	Column	Error	Truncation	Description
Teradata Destina...		Fail component	Fail component	Insert

Set this value to selected cells: Fail component

Figure 38: Teradata Destination Component – Error Output

Options

Error behavior

Select how the Teradata source should handle errors in a flow: ignore the failure, redirect the row, or fail the component.

Truncation

Select how the Teradata source should handle truncation in a flow: ignore the failure, redirect the row, or fail the component.

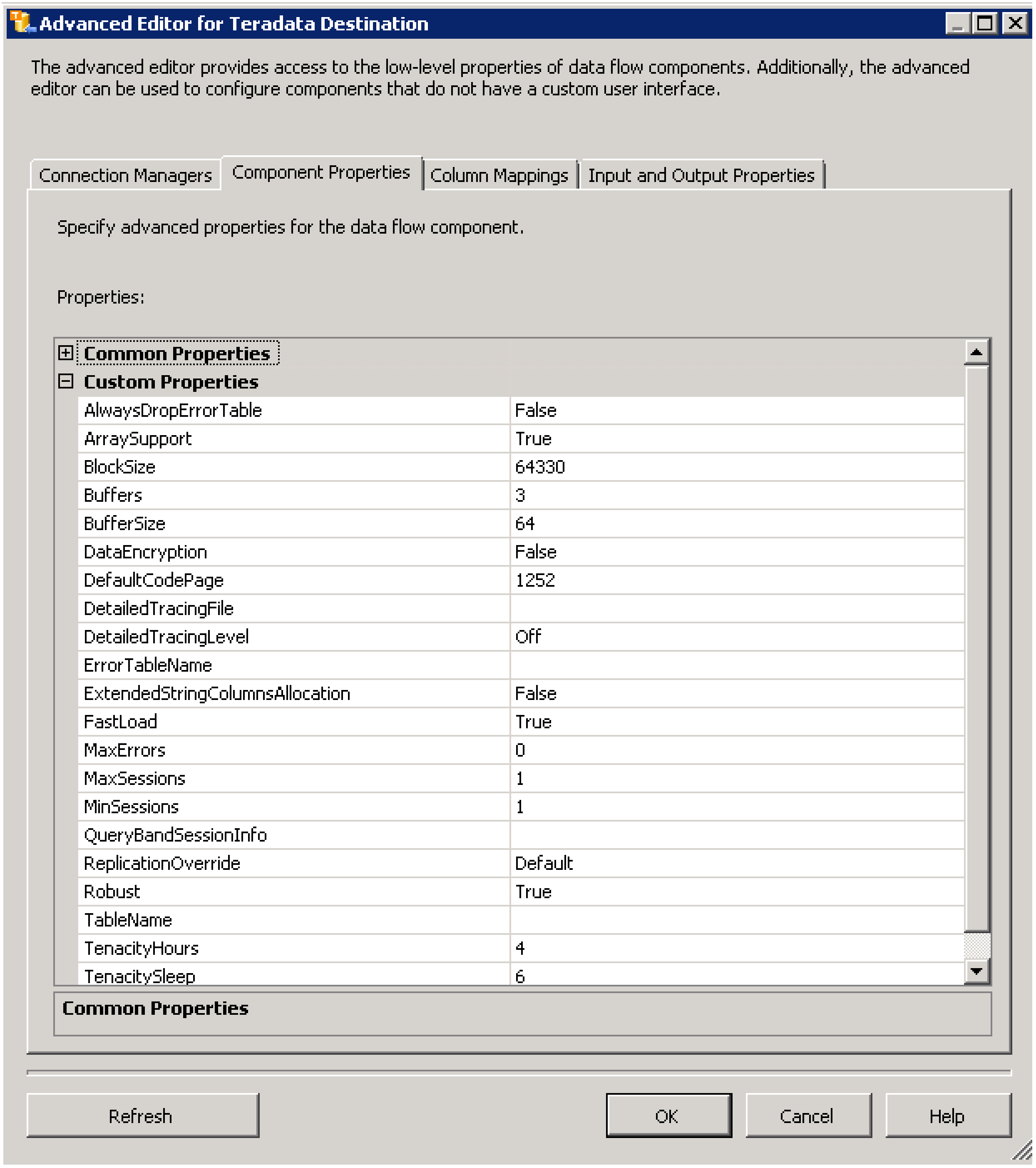
Error-Handling Options

You use the following options to configure how the Teradata source handles errors and truncations:

> **Fail Component**
>
> The Data Flow task fails if an error or a truncation occurs. This is the default behavior.
>
> **Ignore Failure**
>
> The error or the truncation is ignored and the data row is directed to the Teradata source output.
>
> **Redirect Flow**
>
> The error or the truncation data row is directed to the error output of the Teradata source. In this case the Teradata source error handling is used.

Teradata Destination Advanced Editor

The Advanced Editor contains the properties that can be set programmatically. To open the Advanced Editor, in the **Data Flow** screen of your SQL Server Integration Services project, right-click the Teradata destination and then click **Show Advanced Editor**.


Advanced Editor for Teradata Destination

The advanced editor provides access to the low-level properties of data flow components. Additionally, the advanced editor can be used to configure components that do not have a custom user interface.

Specify advanced properties for the data flow component.

Properties:

⊞ **Common Properties**

⊟ **Custom Properties**

AlwaysDropErrorTable	False
ArraySupport	True
BlockSize	64330
Buffers	3
BufferSize	64
DataEncryption	False
DefaultCodePage	1252
DetailedTracingFile	
DetailedTracingLevel	Off
ErrorTableName	
ExtendedStringColumnsAllocation	False
FastLoad	True
MaxErrors	0
MaxSessions	1
MinSessions	1
QueryBandSessionInfo	
ReplicationOverride	Default
Robust	True
TableName	
TenacityHours	4
TenacitySleep	6

Common Properties

Figure 39: Teradata Destination Advanced Editor – Component Properties

Teradata Destination Custom Properties

The following table describes the custom properties of the Teradata destination. All properties are read/write.

Property name	Data type	Description
AlwaysDropErrorTable	Boolean	A value that indicates whether all error tables are dropped even if the Teradata destination fails to read the data. The default value is **False**.
ArraySupport	Boolean	Specifies default array support option for all DMLGroups. The default value is **True** **Note**: This property is available in the Advanced Editor.
BlockSize	Integer	The block size, in bytes, used when returning data to the client. The minimum value is 256 bytes. The default and maximum value is 64,330 bytes.
Buffers	Integer	This specifies whether to increase the number of request buffers. The range of values has a lower limit of two and no upper limit. The default value is **3**. **Note**: This property is available in the Advanced Editor.
BufferSize	Integer	This allows setting the output buffer size (1-64) in kilobytes, used for sending parcels to the Teradata Database. The default value is **64** (i.e. 64260). **Note**: This property is available in the Advanced Editor.
DataEncryption	Boolean	Indicates whether full security encryption of SQL requests, responses, and data is used. • If this property is not selected, no encryption is used. **This is the default value.** • If this property is selected, all SQL requests, responses, and data are encrypted.
DefaultCodePage	Integer	The code page to use if code page information is unavailable from the data source. **Note**: This property is available in the Advanced Editor.
DetailedTracingFile	String	The path that indicates the physical location of the log file. A log file is generated automatically if any **DetailedTracingLevel** value (except **Off**) is selected. **Note**: This property is available in the Advanced Editor.

Property name	Data type	Description
DetailedTracingLevel	Integer	This allows setting the TPT API tracing to different levels. The default value is **Off** **Note**: This property is available in the Advanced Editor.
ErrorTableName	String	The name indicator used to create names for the generated error tables. The default value is the target table name.
ExtendedString ColumnsAllocation	Boolean	A value that indicates whether the Maximal Transfer Character Allocation Factor is used. This value should be set to **True** if the Teradata database **Export Width Table ID** property is set to **Maximal Defaults**. The default value is **False**. **Note**: This property is available in the Advanced Editor.
FastLoad	Boolean	A value that indicates whether fast loading is used. The default value is **True**. This property can also be set in the Teradata Destination Editor (**Connection Manager** page).
MaxErrors	Integer	The maximum number of errors that can be returned before the data flow is stopped. By default the value is 0, which indicates that there is no error limit. If you select **Redirect flow** in the **Error Output** page, all errors that are returned before the data flow reaches the error limit are returned in the error output.
MaxSessions	Integer	The maximum number of sessions that are logged on. This value must be greater than zero. The default value is **1** session.
MinSessions	Integer	The minimum number of sessions that are logged on. This value must be greater than zero. The default value is **1** session.

Property name	Data type	Description
QueryBandSessionInfo	String	Enables a user-defined query band expression that is set for every SQL session connected by the Teradata PT operator. This allows charge back, monitoring, and governance. **Note**: This property is available in the Advanced Editor.
ReplicationOverride	Boolean	The minimum number of sessions that are logged on. This value must be greater than one. The default value is one session for each available AMP. **Note**: This property is available in the Advanced Editor.
Robust	Boolean	This specifies whether or not to use robust restart logic for recovery and restart operations. The default value is **True** **Note**: This property is available in the Advanced Editor.
TableName	String	The name of the table with the data that is being used.
TenacityHours	Integer	The number of hours the driver attempts to log on when the maximum number offload/export operations is already running. The default is 4 hours. **Note**: This property is available in the Advanced Editor.
TenacitySleep	Integer	Specifies the number of minutes the driver pauses before attempting to log on under the restraints defined by the **MaxSessions** and **TenacityHours** properties. The default is 6 minutes. **Note**: This property is available in the Advanced Editor.

Appendix D – Troubleshooting Run-Time Failures

This step-by-step appendix describes how to troubleshoot and analyze run-time failures related to the Microsoft Connector for Teradata by Attunity.

The debugging process depends on the logging facility that SQL Server Integration Services provides for the external providers. Using the verbose log files is necessary if the other debugging facilities of SQL Server Data Tools or Business Intelligence Development Studio did not help, or if the nature of the problems is related to the Microsoft Connector for Teradata, by Attunity.

STEP 1– Eliminate the Common Problems

Check for the common problems and error messages, which you can find in the Event Viewer.

PROBLEM 1 - Permission problems are encountered at run time.

<u>SOLUTION</u>

Check the relevant error message and confirm that the SQL Server service and SQL Server Agent have the required permissions in the specified account.

PROBLEM 2 -A package that runs successfully on a 32-bit platform does not run successfully on a 64-bit platform.

<u>SOLUTION</u>

Make sure you are using the 64-bit connector.

If you are calling DTS packages using the Execute DTS 2000 Package task to run a SQL Server 2000 DTS package, you must run the package in 32-bit mode.

PROBLEM 3 - You encounter general errors in the Event Viewer.

<u>SOLUTION</u>

Always check the Windows Event Viewer for general error messages. If the problem is indeed related to the Microsoft Connector for Teradata, by Attunity, proceed to step 2.

PROBLEM 4 – You encounter TPT registry errors

The Teradata TPT registry key cannot be opened. Verify that the TPT API 12.0 or 13.0 Edition 2 (13.0.0.2) for Windows x86 is installed properly

For more details see:

http://www.attunity.com/forums/microsoft-ssis-teradata-connector/attunity-connector-teradata-version-2-0-a-2852.html

<u>SOLUTION</u>

Install version 13.10 of TPT on your machine.

PROBLEM 5 – Connector supports Teradata TIME data type but not TIME(0)

SSIS default mapping for Teradata TIME data type for TIME(1-6) as DT_STR, but for TIME(0) SSIS maps it to DT_R8 which is incorrect. This returns the following error

Data Type conversion of input column (219) is not supported.

For more details see:

http://www.attunity.com/forums/micorosft-ssis-teradata-connector/loading-data-into-tera-data-time-2847.html

SOLUTION

Need SSIS to map Teradata TIME(0) data type to DT_STR like it does for TIME(1-6).

Will be addressed in Connector version 3.0 for Teradata.

PROBLEM 6 – You encounter a TPT Import error.

[Teradata Destination [987]] Error: TPT Import error encountered during Initiate phase. [SSIS.Pipeline] Error: component "Teradata Destination" (987) failed the pre-execute phase and returned error code 0x80004005.

Trace file states the following:

```
PC_ISSUEDBCHQE: using TDP ID for this query
PC_ISSUEDBCHQE: my_cli->TdpId: '10.1.2.170'
PC_ISSUEDBCHQE: calling DBCHQE for TDP: 'xxx.xxx.xxx.xxx'
PC_ISSUEDBCHQE: length of TDPid is:     10
PC_ISSUEDBCHQE: DBCHQE returns: 207
PC_SETCODE: entering
PC_SETCODE: current condition code: 0
PC_SETCODE: setting condition code: 12
PC_SETCODE: leaving
PC_ISSUEDBCHQE: leaving with Supported flag: 0
PC_ISSUEDBCHQE: leaving with return code:    207
PC_SETCODE: entering
PC_SETCODE: current condition code: 12
PC_SETCODE: leaving
PC_GETDBSLIMITS: leaving with return code: 207
PC_INITCLI: leaving with return code: 207
PC_ISRETRY: entering with result, DBSError: 207, 0
```

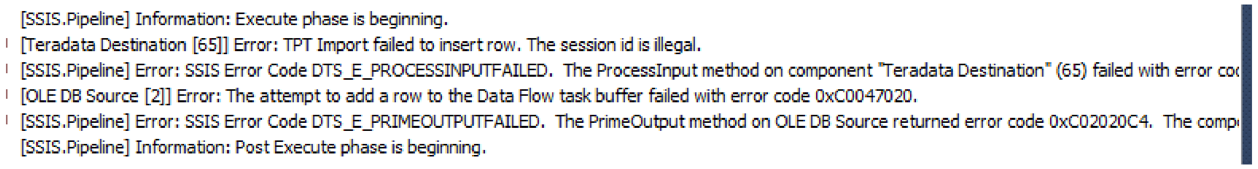
Which translates to, TPT Load operator received a CLI 207 error when trying to connect to the 'xxx.xxx.xxx.xxx' DBS.

SOLUTION

Verify if the local host file has the correct IP address entry that can reach the Teradata system.

PROBLEM 7 – TPT Stream job error

Connector integration with TPT Stream does not incorporate ‘AddSerializeOn’ feature. Hence, if you are trying to load to a NUPI table you might experience the following error or job may appear to hang or is very slow. Blocking or even deadlock may be occurring if the table is a NUPI with duplicates.



```
[SSIS.Pipeline] Information: Execute phase is beginning.
[Teradata Destination [65]] Error: TPT Import failed to insert row. The session id is illegal.
[SSIS.Pipeline] Error: SSIS Error Code DTS_E_PROCESSINPUTFAILED.  The ProcessInput method on component "Teradata Destination" (65) failed with error co
[OLE DB Source [2]] Error: The attempt to add a row to the Data Flow task buffer failed with error code 0xC0047020.
[SSIS.Pipeline] Error: SSIS Error Code DTS_E_PRIMEOUTPUTFAILED.  The PrimeOutput method on OLE DB Source returned error code 0xC02020C4.  The comp
[SSIS.Pipeline] Information: Post Execute phase is beginning.
```

<u>SOLUTION</u>

Reduce number of sessions to 1 or a few or redesign table with UPI.

STEP 2 - Using the Logging Facility

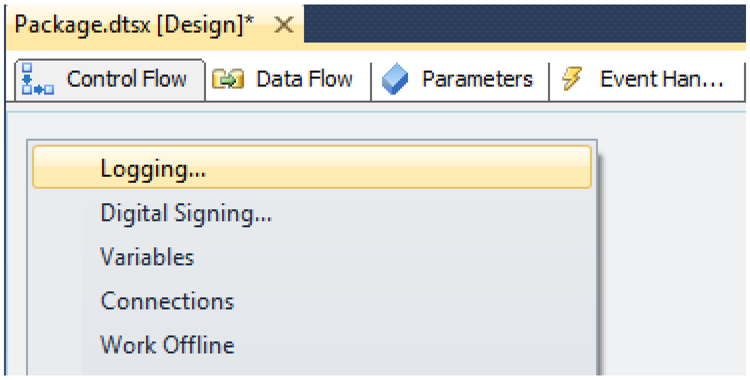
The Microsoft Connector for Teradata outputs meaningful error messages to SQL Server Integration Services; however, there are cases where there is a need for verbose debugging log file, which can show the complete lifecycle of the interaction with the connector and the back-end database.

In these cases SQL Server Integration Services provides a complete logging facility and several logging providers.

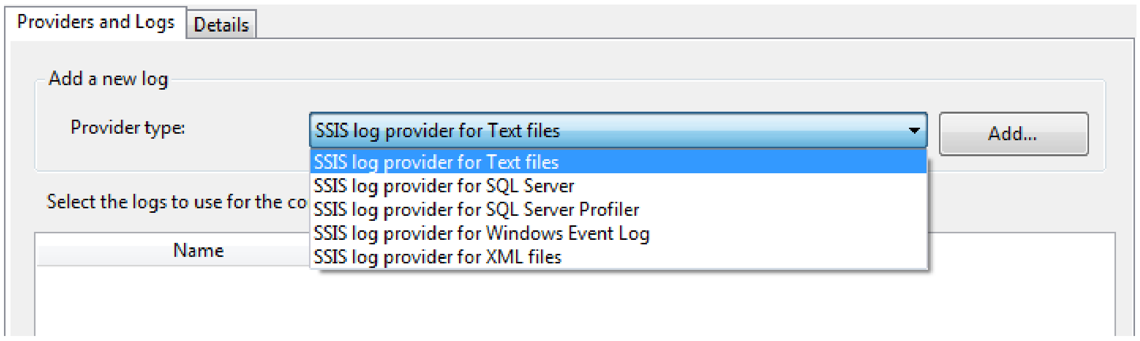
To enable logging, perform the following steps

1. In the SQL Server Data Tools or Business Intelligence Development Studio, open the package for which you want to enable logging.

2. Right-click the **Control Flow** tab, and then click **Logging**.



3. On the **Providers and Logs** tab, click **Add**, and then under **Add a new log**, select a logging provider. For example, to export the log to a simple text file in your file system, click **SSIS log provider for Text files**.



Provider type: SSIS log provider for Text files

Select the logs to use for the co

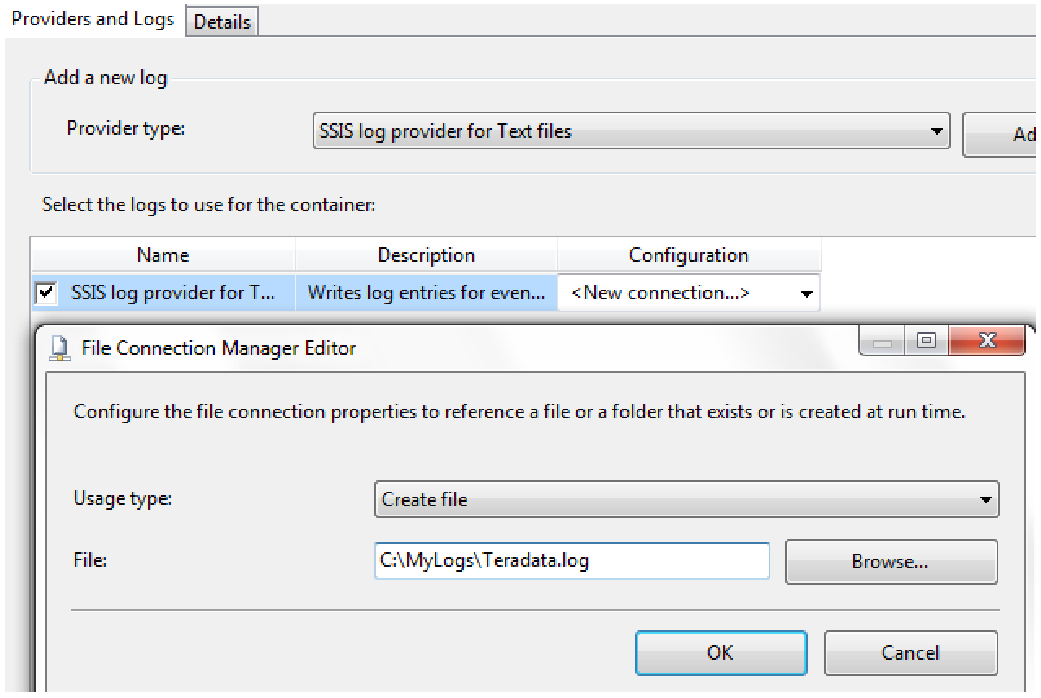
- SSIS log provider for Text files
- SSIS log provider for SQL Server
- SSIS log provider for SQL Server Profiler
- SSIS log provider for Windows Event Log
- SSIS log provider for XML files

To add the selected provider, click **Add**.

To enable the provider, select the check box next to the provider name.

4. To configure the logging provider, click the **Configuration** column. You can either create a new connection or use an existing one.

For the Text provider, you can choose to create a new text file and output the logging to it. You can also append to an existing file.



Provider type: SSIS log provider for Text files

Select the logs to use for the container:

Name	Description	Configuration
✓ SSIS log provider for T...	Writes log entries for even...	<New connection...>

Configure the file connection properties to reference a file or a folder that exists or is created at run time.

Usage type: Create file

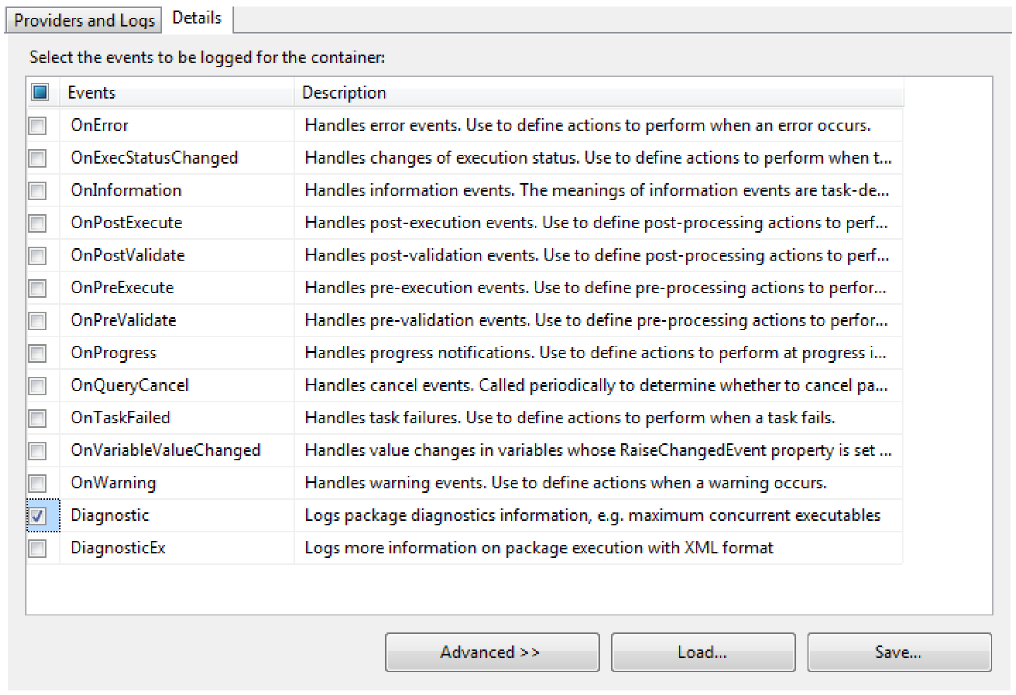
File: C:\MyLogs\Teradata.log

You can define several log file providers to output to multiple log files of different formats.

5. Select the diagnostic level for debugging. Click the **Details** tab, and then select the events you want to log.

6. For the Teradata Connector, click **Diagnostic**, which will log among other things the important interactions with the Teradata Parallel Transporter API interface of Teradata.

   Note: You can select other events to be logged, such as **OnError**, **OnInformation**, or **OnWarning**.



Select the events to be logged for the container:

☐	Events	Description
☐	OnError	Handles error events. Use to define actions to perform when an error occurs.
☐	OnExecStatusChanged	Handles changes of execution status. Use to define actions to perform when t...
☐	OnInformation	Handles information events. The meanings of information events are task-de...
☐	OnPostExecute	Handles post-execution events. Use to define post-processing actions to perf...
☐	OnPostValidate	Handles post-validation events. Use to define post-processing actions to perf...
☐	OnPreExecute	Handles pre-execution events. Use to define pre-processing actions to perfor...
☐	OnPreValidate	Handles pre-validation events. Use to define pre-processing actions to perfor...
☐	OnProgress	Handles progress notifications. Use to define actions to perform at progress i...
☐	OnQueryCancel	Handles cancel events. Called periodically to determine whether to cancel pa...
☐	OnTaskFailed	Handles task failures. Use to define actions to perform when a task fails.
☐	OnVariableValueChanged	Handles value changes in variables whose RaiseChangedEvent property is set ...
☐	OnWarning	Handles warning events. Use to define actions when a warning occurs.
✓	Diagnostic	Logs package diagnostics information, e.g. maximum concurrent executables
☐	DiagnosticEx	Logs more information on package execution with XML format

7. To save the current configuration, click **Save**, and then click **OK**.

8. To save the changes to the package, on the **File** menu, click **Save Selected Items**.

9. Execute the problematic package or the package that you want to debug, and then review the output log file.

STEP 3 - Understanding the Log File

The verbose log file contains the details and interaction of the different components in your package. Quickly reviewing the log file can reveal problems that can be fixed without the need to involve technical support.

Many general problems related to SQL query, the backend database, or SQL Server Integration Services can be solved by reviewing the log file. If for any reason you cannot understand the cause of the failure, you should make a support call to get an explanation of the failure.

Additional Information

In addition to the 3 steps above, SQL Server 2012 introduces features and functionalities that can make troubleshooting of SSIS easier.

This article presents additional tools and techniques for troubleshooting during development; while this other one presents tools and techniques for troubleshooting during execution.