IBM InfoSphere DataStage and QualityStage
Version 9 Release 1

Connectivity Guide for Netezza Performance Server

IBM
Note

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 65.
Chapter 1. Configuring access to Netezza databases

To configure access to Netezza databases, you must install database client libraries and include the path to these installed libraries in the library path environment variable. For more information about setting environment variables, see the topic about setting environment variables.

Procedure
1. Install database client libraries.
2. Configure access to Netezza databases.

Configuring access to Netezza databases

To configure access to Netezza database, you must install and configure the Netezza ODBC driver and create the data source.

Configuring access to Netezza databases on Linux and UNIX

To configure access to Netezza database you must specify parameters in the .odbcinst.ini file to configure the Netezza ODBC driver and also modify the .odbc.ini file to configure the data sources.

Before you begin
- Install client libraries.

About this task

If an .odbcinst.ini configuration file exists, you can modify the same file. If there is no existing .odbcinst.ini configuration file, then you can use the odbcinst.ini.sample to create the .odbcinst.ini configuration file. In most scenarios, you can use the contents of the odbcinst.ini.sample file without any changes. However, in the following scenarios, you must change the configuration file:
- If your client system was configured for ODBC drivers other than the Netezza ODBC driver and you want to continue to use those ODBC drivers, do not modify the existing entries in the .odbcinst.ini file. Add an entry for the Netezza ODBC driver at the end of the existing contents of the .odbcinst.ini file.
- If the Netezza client software and a Netezza ODBC driver were installed on your client system, check if the Netezza ODBC driver is configured. If it is not, add an entry to the end of the existing contents of the .odbcinst.ini file.

If the .odbc.ini configuration file exists in your home directory (For example, /home/myname) check if it contains entries for the Netezza appliance data sources to access. If it does not, copy the contents of the odbc.ini.sample file to the end of your existing .odbc.ini configuration file. Do not modify any existing entries in the file.

If you are using the InfoSphere® Information Server version of the .odbc.ini configuration file on Linux, create a symbolic link in the folder where the configuration file exists to make sure that the Netezza connector works correctly:
1. Log on as the InfoSphere DataStage® administrator.
2. To change to the installation directory of InfoSphere Information Server, enter the command: `cd /opt/IBM/InformationServer/Server/DSEngine`.

3. To create a symbolic link, enter the command: `ln -s .odbc.ini odbc.ini`.

**Procedure**

1. Log in using your user ID and password.

2. Configure the Netezza ODBC driver.
   
   
   b. Modify the configuration entries depending on your requirement. Consult your Netezza system administrator to check if you must modify any specific configuration entries for your installation.
   
   c. Save the file as `.odbcinst.ini`.

3. Configure the Netezza appliance data sources.
   
   a. Copy the contents of `.odbc.ini.sample` file into your home directory (For example, `/home/mynname`) and rename it `.odbc.ini`.
   
   b. Optional: To add the Netezza data sources to an existing `.odbc.ini` file, add the lines after `[NZSQL]` from the sample file to the existing `.odbc.ini` file. In the [ODBC Data Sources] section, add NZSQL = NetezzaSQL to the list of data source names.
   
   c. Save and close the file.

4. Set the following environment variables:
   
   ```
   export ODBCINI=path_to_odbc.ini_file
   export NZ_ODBC_INI_PATH=location_of_odbc.ini_file
   ```

   **Note:** If the Netezza entries were added to an existing `odbc.ini` file, set only the `NZ_ODBC_INI_PATH` variable.

5. To restart the server engine and the ASB Agent, enter the following command.
   
   ```
   cd Install_directory/Server/DSEngine/bin
   ./uv -admin -stop
   ./uv -admin -start
   cd Install_directory/ASBNode/bin
   ./NodeAgents_env_DS.sh
   ./NodeAgents.sh stopAgent
   ./NodeAgents.sh start
   ```

**Configuring access to Netezza databases on Microsoft Windows**

If InfoSphere Information Server runs on the Microsoft Windows operating system, you must create and configure the data source after you install the Netezza ODBC driver.

**Before you begin**

- Install database client libraries.
- On 64-bit Windows computers, make sure that you run the 32-bit version of the Microsoft ODBC Data Source Administrator `C:\Windows\SysWOW64\odbcad32.exe`, as InfoSphere Information Server is a 32-bit application. If you run the 64-bit version of the ODBC administrator application, the Netezza connector fails to locate the specified data source name. If the ODBC administrator application is not accessible through the File menu by default, use the Windows Explorer to access the application.
  - On 32-bit Windows, the 32-bit driver is installed in the `C:\Windows\System32` directory.
– On 64-bit Windows, you can install both 32-bit and 64-bit drivers. The 32-bit driver is installed in the C:\Windows\SysWOW64 directory and 64-bit version is installed in the C:\Windows\System32 directory.

**Procedure**

1. To create the data source:
   a. Do one of the following actions depending on your Operating System:
      - On a 32-bit Windows system, click Start > Control panel > Administrative Tools > Data Sources (ODBC).
      - On a 64-bit Windows system, use Explorer to navigate to C:\Windows\SysWOW64\odbcad32.exe.
   b. On the System DSN page, click Add.
   c. On the Create New Data Source page, select NetezzaSQL as the driver to set up the data source for, and then click Finish.

2. To configure the ODBC driver:
   a. On the Netezza ODBC Driver Setup page, specify details about the data source.
   b. In the Server field, specify the host name or the IP address of the Netezza system to which the ODBC driver connects.
   c. To test the connection, specify the username and password, and then click Test Connection.

**Setting the library path environment variable**

To apply an environment variable to all jobs in a project, define the environment variable in the InfoSphere DataStage and QualityStage® Administrator. The values that are specified for the library path and path environment variables at the project or job level are appended to the existing system values for these variables.

**About this task**

For example, suppose that directory /opt/branded_odbc/lib is specified as the value for the library path environment variable at the project level. Directory /opt/IBM/InformationServer/Server/branded_odbc/lib, which contains the same libraries but in a different location is already in the library path that is defined at the operating system level or the dsenv script. In this case, the libraries from directory /opt/IBM/InformationServer/Server/branded_odbc/lib are loaded when the job runs because this directory appears before directory /opt/branded_odbc/lib in the values that are defined for the library path environment variable.

The name of the library path environment variable depends on your operating system.

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Library path environment variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>PATH</td>
</tr>
<tr>
<td>HP-UX</td>
<td>SHLIB_PATH</td>
</tr>
<tr>
<td>IBM® AIX®</td>
<td>LIBPATH</td>
</tr>
<tr>
<td>Other supported Linux and UNIX</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>systems, and HP-IA</td>
<td></td>
</tr>
</tbody>
</table>
On Linux or UNIX operating systems, the environment variables can be specified in the dsenv script. InfoSphere Information Server installations on Windows operating system do not include the dsenv script.

**Setting the library path environment variable in the dsenv file**

On Linux or UNIX operating systems, you can specify the library path environment variables in the dsenv script. When environment variables are specified in the dsenv script, they apply to all InfoSphere DataStage projects that run under the InfoSphere Information Server engine.

**Before you begin**

Install the client libraries.

**Procedure**

1. Log in as the root user.
2. Back up the $ISHOME/Server/DSEngine/dsenv script. $ISHOME is the InfoSphere Information Server installation directory (/opt/IBM/InformationServer by default).
3. Open the dsenv script.
4. Add the path to the directory that contains the client libraries to the library path environment variable.
5. To stop and start the InfoSphere Information Server engine, enter the following commands:
   
   ```
   bin/uv –admin –stop
   bin/uv –admin –start
   ```
6. To change directory to the ASB Agent home directory, enter the following commands:
   
   ```
   cd Install_directory/ASBNode/bin
   ```
7. To stop and start the ASB Agent processes, enter the following commands:
   
   ```
   ./NodeAgents.sh stopAgent
   ./NodeAgents.sh start
   ```

**Results**

After you restart the ASB Agent process, the InfoSphere Information Server domain services (WebSphere® Application Server) take approximately a minute to register the event.

**Setting the library path environment variable in Windows**

On the Windows operating system, both the library path and PATH environment variables are represented by the PATH. For InfoSphere Information Server engine and ASB Agent processes to detect changes in the environment variables, the changes must be made at the system level and the InfoSphere Information Server engine must be restarted.

**Before you begin**

Install the client libraries.

**Procedure**

1. To edit the PATH system environment variable, click **Environment Variable** in **Advance System Settings**, and then select PATH.
2. Click **Edit**, then specify the path to the directory containing the client libraries.
3. Click OK.
4. Restart the InfoSphere Information Server engine.
5. Restart the ASB Agent processes.

---

**Testing database connections by using the ISA Lite tool**

After you establish connection to the databases, test the database connection by running the IBM Support Assistant (ISA) Lite for InfoSphere Information Server tool.

For more information about the ISA Lite tool, see the topic about installation verification and troubleshooting.
Chapter 2. Netezza enterprise stage

Netezza enterprise stage is a database stage. Using Netezza enterprise stage, you can write bulk data to Netezza Performance Server 8000. This topic provides a brief introduction to Netezza Performance Server, describes how Netezza enterprise stage works, and explains how to use the stage to write bulk data to Netezza Performance Server.

About Netezza Performance Server 8000

Netezza Performance Server 8000 is an enterprise-class database appliance. Below are the major features and benefits of Netezza Performance Server:

- Netezza Performance Server is specifically designed to meet the performance requirements of business intelligence (BI) and business analytics for Terascale databases in real time.
- Using Netezza Performance Server, you can leverage the availability of open standards and inexpensive components.
- Netezza Performance Server supports BI with standard SQL queries, and is therefore easy to use. You can run your own ad hoc queries, and you do not need skilled database administrators to tune the database for all possible queries.

To leverage Netezza Performance Server, use the available load methods and supported client interfaces.

Load methods

Two alternative methods for writing data to Netezza Performance Server are available to you - external table and nzload. Listed below are the major differences between these two load methods:

<table>
<thead>
<tr>
<th>Table 1. Load methods in Netezza enterprise stage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>External Table</strong></td>
</tr>
<tr>
<td>An external table is a temporary table that Netezza enterprise stage creates in the database. Such a table contains data in the form of delimited characters, the delimiter being comma. If you choose this load method, data from a data source is loaded to the destination table in Netezza Performance Server using an external table.</td>
</tr>
<tr>
<td>Use the external table load method when the source table columns have default values and data encoding is in variable format, such as UTF-8 and LATIN9.</td>
</tr>
</tbody>
</table>
Table 1. Load methods in Netezza enterprise stage (continued)

<table>
<thead>
<tr>
<th>External Table</th>
<th>nzload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the CREATE EXTERNAL TABLE command, you can write data to or remove data from a Netezza database from or into an external source that is not part of your database. The external table load method functions using 3 commands:</td>
<td>The nzload command is an ASCII, bulk data loader designed to load data quickly while simultaneously culling erroneous or ambiguous content. The nzload command requires that you specify the textual format and enforces consistency.</td>
</tr>
<tr>
<td>• CREATE TABLE: Supports default values for each column in the source table.</td>
<td>The nzload command, although driven by the target table's schema, ignores all default value specifications.</td>
</tr>
<tr>
<td>• SELECT: Reads data from an external table.</td>
<td></td>
</tr>
<tr>
<td>• INSERT: Interprets textual values and decides which input format to use.</td>
<td></td>
</tr>
<tr>
<td>The external table load method uses the INSERT statement to load data. To execute the INSERT command, you only require DML privileges (CREATE TABLE, SELECT, and INSERT privileges).</td>
<td>To use the nzload load method, you require CREATE TABLE and LOAD privileges for the destination table.</td>
</tr>
<tr>
<td>For the external table load method, the log files are created in the ../tmp directory in the Netezza Performance Server system.</td>
<td>For the nzload load method, the log files are created in the ../tmp directory in the client system.</td>
</tr>
</tbody>
</table>

Note: When using the nzload load method, ensure that the source schema order is identical with the schema of the destination table.

Client interfaces for accessing Netezza Performance Server

You can use the client interfaces listed below to access Netezza Performance Server. IBM InfoSphere DataStage only supports client interfaces corresponding to Netezza server, for example, ODBC driver version 2.05. ODBC driver version 2.05 is in conformance with ODBC 3.0 specifications.

Note: The ODBC driver is required for accessing Netezza Performance Server.

• Clients required for accessing Netezza Performance Server:
  - **ODBC client interface**: To access Netezza Performance Server, you must install the ODBC client interface. The ODBC client allows applications and tools for building applications, to access SQL databases.
  - **Netezza client interface**: If you want to load data to Netezza Performance Server by using the nzload load method, then you must install the Netezza client.

• Optional clients for accessing Netezza Performance Server:
  - **Microsoft Windows client interface**: The Windows client consists of NzAdmin, a GUI-based tool. You can use the Windows client interface to run system and database operations.
  - **Web client interface**: The Web client consists of Webadmin, a browser-based administrator tool. You can use the Web client interface to monitor and administer an Netezza Performance Server system from popular Web browsers.
  - **Linux or Solaris client or command line interface**: You can use the Linux or Solaris client to enter SQL statements. The Linux or Solaris client or command line interface also provides a view of returned result sets from a Linux or
Solaris client system and the Netezza command line interface commands, using which you can manage the system and databases.

- **JDBC client interface:** The JDBC client allows Java™ programs to access SQL databases.

**Supported operating systems or platforms**

You can connect to Netezza Performance Server from systems that are running the below platforms:

- IBM AIX 6.1
- HP-UX (PA-RISC) 11.0
- Red Hat Linux AS 3.0
- SunOS 5.10
- Microsoft Windows 2008 Server and above

**How Netezza enterprise stage works**

Here is an overview of how Netezza enterprise stage works. Netezza enterprise stage is a read and write stage.

The **write** stage takes bulk data from a data source and writes that data to a specified destination table in Netezza Performance Server. The input data source for Netezza enterprise stage can be another database stage, a generated file, and so on. You can write data to Netezza Performance Server using any of the two available load methods - external table or nzload. For detailed information about the load methods, see the Load methods section. The forthcoming sections will describe in detail how to select a load method.

The graphics below show the workflows of Netezza enterprise stage for external table and nzload load methods respectively:
The read stage locates bulk data from a specified target table in Netezza Performance Server and writes the data in to a data source or another database stage.

**Working with Netezza enterprise stage**

Below are the prerequisites for using Netezza enterprise stage. You will also learn how to add a Netezza enterprise stage to your job.

**Prerequisites**

To write data to Netezza Performance Server using Netezza enterprise stage, you must install the required components. Then you must configure the stage and your system appropriately.

**Installing and configuring required components**

**About this task**

Listed below are the installation and configuration requirements:

**Procedure**

1. Install Netezza server and client. Netezza enterprise stage supports Netezza server 2.05 Patch 4 and higher. You must install the Netezza client if you want to use the nzload load method.
2. Install and configure the nzload utility and ODBC driver provided by Netezza Performance Server. Netezza enterprise stage uses this ODBC driver to retrieve metadata for the destination table or file. Ensure that you install the 2.05 version of this ODBC driver. This driver is in conformance with 3.0 ODBC specifications.
3. Ensure that the DataDirect's ODBC driver manager is installed and configured on UNIX systems.
4. Obtain explicit LOAD privileges for the target table in Netezza Performance Server.

**Note:** For detailed information or help for installing and configuring the above, refer to the documentation that accompanies the software.

5. Create the odbc.ini file in $DSHOME. To create the odbc.ini file, you must first configure the .odbc.ini file located in $DSHOME, by adding necessary information such as database name, host name, user name, and password. For
detailed information about configuring the .odbc.ini file, refer to the
Configuring data sources on a UNIX platform section. Then copy the contents
of the .odbc.ini file to the odbc.ini file in the same location. Alternatively, you
can create a soft link to the .odbc.ini file.

6. Set user defined and general environment variables appropriately. The table
below provides information about how to set user defined and general
environment variables.

<table>
<thead>
<tr>
<th>Environment variable</th>
<th>Type</th>
<th>Settings</th>
<th>Description/Purpose</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETEZZA</td>
<td>User defined</td>
<td>$NETEZZA</td>
<td>Specifies the Netezza home directory</td>
<td>$export NETEZZA=/disk3/Netezza</td>
</tr>
<tr>
<td>NZ_ODBC_INI_PATH</td>
<td>User defined</td>
<td>$NZ_ODBC_INI_PATH</td>
<td>Points to the location of the .odbc.ini file. This is specifically required for UNIX systems.</td>
<td>$DSHOME/odbc.ini</td>
</tr>
<tr>
<td>APT_DEBUG_MODULE_NAMES</td>
<td>User defined</td>
<td>$APT_DEBUG_MODULE_NAMES</td>
<td>Prints debug messages from a specific module</td>
<td>odbcstmt odbcenv nzetwriteop, nzutils, nzwriterep, nzetsubop</td>
</tr>
<tr>
<td>PATH</td>
<td>General</td>
<td>NETEZZA/bin</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>LIBPATH/LD_LIBRARY_PATH/SHLIB_PATH</td>
<td>General</td>
<td>NETEZZA/lib or NETEZZA/lib64</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>APT_NETEZZA_LOAD_FILES</td>
<td>User defined</td>
<td>$APT_NETEZZA_LOAD_FILES</td>
<td>Location of Netezza log/error files</td>
<td>/home/dsadm/nzlogs</td>
</tr>
<tr>
<td>APT_TMPDIR</td>
<td>User defined</td>
<td>$APT_TMPDIR</td>
<td>Location of Netezza log/error files</td>
<td>/home/dsadm/nzlogs</td>
</tr>
<tr>
<td>APT_NZ_UNIQUE_KEY_NOT_ALLOWED</td>
<td>User defined</td>
<td>$APT_NZ_UNIQUE_KEY_NOT_ALLOWED</td>
<td>overrides Unique key specified in the graphical user interface</td>
<td>APT_NZ_UNIQUE_KEY_NOT_ALLOWED=1</td>
</tr>
</tbody>
</table>

Results

You must set PATH / LIBPATH / LD_LIBRARY_PATH / SHLIB_PATH based on the platform operating on your system. Use information provided in the below table to set PATH / LIBPATH / LD_LIBRARY_PATH / SHLIB_PATH.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Environment variable to be set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>PATH</td>
</tr>
<tr>
<td>IBM AIX</td>
<td>LIBPATH</td>
</tr>
<tr>
<td>Solaris, Linux</td>
<td>LD_LIBRARY_PATH</td>
</tr>
<tr>
<td>HP-UX</td>
<td>SHLIB_PATH</td>
</tr>
</tbody>
</table>

- Update the .../etc/hosts file after installing the Netezza client. You must update this file with the Netezza server and IP address information. To access this file using the command prompt, use the etc/hosts command line. In Windows machines, this file is located in Windows/system32/drivers/etc.
Note: You can choose to configure Netezza enterprise stage from the IBM InfoSphere DataStage palette or by using the command prompt. This topic only describes how to configure the stage by using the InfoSphere DataStage palette. For details about configuring the stage using the command prompt, see IBM InfoSphere DataStage and QualityStage Parallel Job Advanced Developer's Guide.

Configuring data sources
About this task

To be able to work with Netezza enterprise stage, you must configure data sources appropriately. Here you will learn how to configure data sources for Microsoft Windows and UNIX.

Configuring data sources on a Windows platform:
About this task

From your desktop, click Start > Control Panel > Administrative Tools > Data sources (ODBC).

From the ODBC Data Source Administrator dialog, add a new data source name. To add a new data source name, select the appropriate ODBC driver from the Create New Data Source dialog, and set appropriate parameters.

When setting the parameters, make sure that you clear the ReadOnly option. This option is selected by default.

Configuring data sources on a UNIX platform:
About this task

To configure data source names in UNIX machines for Netezza enterprise stage, you must create and configure the odbc.ini file in $DSHOME. Add the following entries to the odbc.ini file.

```
[ODBC Data Sources]
NZSQL = NetezzaSQL
[NZSQL]
Driver = /usr/local/nz/lib/libnzodbc.so
Description = NetezzaSQL ODBC
Servername = 127.0.0.1
Port = 5480
Database = dev
Username = guest
Password = password
ReadOnly = false
ShowSystemTables = false
LegacySQLTables = false
LoginTimeout = 0
QueryTimeout = 0
DateFormat = 1
NumericAsChar = false
SQLBitOneZero = false
StripCRLF = false
securityLevel = preferredUnSecured
tcaCertFile =
```

Set the user defined and general environment variables appropriately.
Netezza enterprise stage operations

You can use a Netezza enterprise stage to perform database operations that are listed below:

- Read data from Netezza Performance Server.
- Write data to Netezza Performance Server by using nzload or an external table.
- Update/Upsert data to a target table.

Read operation
Use the Netezza enterprise in the read mode to read a table from Netezza Performance Server. The graphic below is a sample InfoSphere DataStage job that includes a Netezza enterprise stage in read mode.

Write operation
Use the Netezza enterprise stage in write mode to perform the following upsert operations: Insert records into an external data source table and delete records from an external data source table.

Update/Delete/Upsert operation
Use the Netezza enterprise stage in upsert mode to perform the upsert operations listed below.

- Update records in an external data source table.
- Delete records from an external data source table.
- Update and then insert records into an external data source table.
- Delete and then insert records into an external data source table.

Writing data to Netezza Performance Server using Netezza enterprise stage

About this task
To write data to Netezza Performance Server, you must create a job using the available drag-and-drop options, and then set appropriate properties for Netezza enterprise stage and its input link. Here you will learn to use the options available in order to write data to Netezza Performance Server.

Setting up the attributes of the Input link
About this task

Netezza enterprise stage has only one input link that writes data to one table.

When you double-click the Netezza enterprise stage icon from the IBM InfoSphere DataStage palette, a dialog is displayed. This dialog displays two tabs: Stage and Input. The Input tab is active by default, and this tab displays the Input page.

You use the Stage page that you see when you click the Stage tab, to set up general attributes for Netezza enterprise stage. For detailed information about setting such attributes, read the Setting attributes for Netezza enterprise stage section.

Using this page, you set the attributes of the input link to determine the way in which data will be written to Netezza Performance Server. You perform the following tasks to set up the input link attributes:

- Specifying a description of Netezza enterprise stage
- Setting up the properties of Netezza enterprise stage
- Specify partitioning information for incoming data
- Specify column definitions for incoming data
- Specify buffering settings for the input link

**Specifying a description of Netezza enterprise stage:**

**About this task**

Use the General tab on the Input page to specify a description of the input link. This is an optional feature.

**Setting up the properties of Netezza enterprise stage:**

**About this task**

On the Input page, the Properties tab is active by default. Under the Properties tab, you see a list of properties distributed under 3 categories, Target, Connection, and Options. To edit a property, select the property from the list. You see information about the selected property in the fields next to the list of properties. Read below to know how to set these properties. Before you understand how to set these properties, you must know about the log files.

**Log files**

To know about errors that occur during any database operations, as well as the success or failure of these operations, you must verify the log files. For ease of locating the log files, the naming convention of log files for external table and nzload methods are shown below:

- Netezza error log files are written to a directory based on the APT_NETEZZA_LOAD_FILES or APT_TMPDIR environment variable that is defined in the order of priority. By default, the error log files are written into .../tmp directory in Netezza Performance Server. The log file names for external table load method are:
  - NZORCH.<timestamp>_<partition number>_<database name>.nzlog and
  - NZORCH.<timestamp>_<partition number>_<database name>.nzbad

- For the nzload load method, the log files are created in the .../tmp directory in the client machine. The log file for nzload load method are:
  - <database name>_<table name>_<timestamp.partition number>.log and
  - <database name>_<table name>_<timestamp.partition number>.bad

**Target**

Under this category, you must specify values for the following options:

- Load Method
- Table
- Write Method
- Write mode

These properties are described below in detail.

- **Load Method**
  Specify whether you wish to write data using the nzload utility or an external table. The default load method is nzload. You can use nzload if you have explicit LOAD privileges for the destination table in Netezza Performance Server.
and if you have installed Netezza client in your system. Otherwise, you should choose ET (external table) as the load method. The ET load method only requires you to have CREATE TABLE, SELECT, and INSERT privileges for the destination table.

**Note:** When you run a job to which a Netezza enterprise stage with ET load method is added, the log file is created in the /tmp directory of Netezza Performance Server, and not in the InfoSphere DataStage server.

**Table**

Specify the name of the table to which you want to write data. Specify the table name in this field using any one of the methods below:

- **Method 1:** Enter the table name in the Table field on the right side of the Properties page.
- **Method 2:** Insert the desired value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then Insert Job Parameters.

A list of available job parameters appears from which you can choose the appropriate parameter. If you wish to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

Using the InfoSphere DataStage and QualityStage client, you can also create parameters at the project level for all jobs within the project.

**Write Method**

Provides an option to either write data to a target table or update/upsert/delete data in a target table.

If you select **Write Method = Write**, you can choose to load data either by native bulk nzload utility or by using external tables.

- **Write Mode:** Specifies how Netezza enterprise stage should write data to Netezza Performance Server. You can choose from the following options:
  - **Append:** Appends all rows.
  - **Create:** Creates a new table.
  - **Replace:** Replaces an existing table or all the rows in an existing table. If the table does not exist, then a new table is created. If the destination table exists, then the table is dropped and a new table is created.
  - **Truncate:** Truncates records in an existing table.

If you select **Write Method = Update**, the **Update Mode** option displays. When you select **Update Mode = Update** or **Update then Insert** or **Delete then Insert**, **Write Mode** is no longer displayed under Target category.

- **Update Mode:** Provides an option to either Update (Update Mode = Update) a record or Update and then Insert (Update Mode = Upsert) a record or Delete a record or Delete then Insert records from the target table based on the combination keys. Specify the following values:
  - **Key Column:** Select the key column name that has to be considered as a key and that is a part of the combination key.

  **Note:** You should select different Key Columns if the combination key has more than one Key Column. If the selected Key Columns are duplicates, the Netezza Performance Server displays an error message and Update will not be done.

- **Unique Key Column:** Select a column as a unique key. A unique key column should be a date/time or timestamp or sequence number (int /
You should specify unique values in the input dataset for all the records. Unique key column should not be a part of the combination key and none of the Key Columns of the combination key should be selected as a Unique Key. You must specify a value for the unique key column. However, you can override this option by using the APT_NZ_UNIQUE_KEY_NOT_ALLOWED environment variable. When it is set, the unique key column specified on the graphical user interface will not be considered while updating. The columns of the composite key only are taken into consideration while updating the table in Netezza database.

- **Updatable Column:** Select the column of the input dataset that requires to be updated.

**Note:** Do not select the columns that are already selected as key or unique key columns.

- **Temporary Work Table:** Specify the temporary work table name. By default, the Temporary Work Table is created by Netezza Enterprise Edition Upsert stage in the Netezza Performance Server.

- **Drop Work Table:** Select True to drop a table and the corresponding columns. Select False to retain the table and the corresponding columns.

**Connection**

Under this category, you must specify the following values:

- **Data source:** Specify the name of the data source.
- **Database:** If using the nzload load method, then specify the name of the database to which you need to connect. Ensure that you enter the same name as the database name specified in the data source. Otherwise, data will not be written. If using the ET load method, then database name is not required. The reason is that ET load method uses the ODBC DataDirect driver.

**Note:** If the database details that you provided are incorrect, you receive an alert.

- **Password:** Specify the password for connecting to the database.
- **Server:** Specify the name of the server that you wish to use for loading data to Netezza Performance Server.
- **Temporary Work table (TWT) in Different Database:** The default value is False. Select True to use a temporary work table (TWT) residing in a database other than the target table database. Note that the databases need to both reside on the same physical Netezza server.
- **TWT Database:** Specify the name of the TWT database that needs to be used.
- **User:** Specify the user name for connecting to the data source.

To specify values for the above fields, use any one of the methods below:

- **Method 1:** Enter the appropriate value in the corresponding field on the right side of the Properties page. For example, when you select Data source, you can directly type the desired data source name in the Data source field that appears on the right side of the page.

- **Method 2:** Insert the desired value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then Insert Job Parameters. A list of available job parameters appears from which you can choose the appropriate parameter. If you wish to create a new parameter for the job, click [New...:] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.
Options

Under this category, you must specify the values for the following options:

- Truncate Column Names
- Open Command
- Close Command
- Drop Unmatched Column
- Delimiter
- Netezza Load Options

Under the Options category, Close Command and Truncate Column Names fields appear by default. You can add the other properties mentioned above from the Available properties to add list that appears in the bottom right side of the Properties page. All the above options are described below.

- **Truncate Column Names**
  
  This is an optional field. Select True to truncate column names to the size allowed by Netezza enterprise stage. Select False to disable truncation of field names.

- **Open Command**
  
  This is an optional field. Specify the appropriate command in single quotes.

  Netezza enterprise stage parses and runs this command on all processing nodes before opening the source table.

  To specify a value for Open Command, use any one of the methods below:

  - **Method 1:** Enter the command in the Open Command field on the right side of the Properties page.
  
  - **Method 2:** Insert the desired value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then Insert Job Parameters.

    A list of available job parameters appears from which you can choose the appropriate parameter. If you wish to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

- **Close Command**
  
  This is an optional field. Specify the appropriate command in single quotes.

  Netezza enterprise stage parses and runs this command on all processing nodes after processing the table.

  To specify a value for Close Command, use any one of the methods below:

  - **Method 1:** Enter the command in the Close Command field on the right side of the Properties page.
  
  - **Method 2:** Insert the desired value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then Insert Job Parameters.

    A list of available job parameters appears from which you can choose the appropriate parameter. If you wish to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

  If the specified close command fails, a fatal error occurs. This error is logged in the appropriate log file in $DSHOME. However, even when your query fails, the status of the job that you are creating is displayed as OK in the log file. You must visit the log file to find out whether your query was successful.
Note: The log files for job operations are located within $DSHOME.

- **Drop Unmatched Column**
  This is an optional field. Select **True** to have Netezza enterprise stage drop all the columns that do not match columns of table that you have chosen to append. You do not receive any alert messages when unmatched columns are dropped. The default value is **False**. If you retain this default value, then at runtime, you receive an alert message and the job terminates.

- **Delimiter**
  Specify in this field the delimiter for field values or columns. Make sure that you specify an ASCII character in this field. The default delimiter is @. You can use any ASCII character except space (" ") and hyphen ("-"). Hyphen is already used as the date/time/timestamp delimiter.

  To specify the delimiter, use any one of the methods below:
  - **Method 1:** Enter the desired ASCII character in the **Delimiter** field on the right side of the Properties page.
  - **Method 2:** Insert the desired value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then **Insert Job Parameters**.

  A list of available job parameters appears from which you can choose the appropriate parameter. If you wish to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

  **Note:** If you choose the ET load method, Netezza enterprise stage creates external tables for each operation performed. If these tables are not found, then an error is logged in the log file and the stage tries to drop the tables that are not found. However, the error that occurs does not stop you from creating the job. Delimiter is maintained in the graphical user interface to support backward compatibility.

- **Netezza Load Options:** Use this option to specify the load options that are not mentioned by default to the ET or nzload load methods. Load Options are directly passed to nzload utility or the ET.

  **Note:** When **Load Method = nzload**, do not use the following load options:
  - -u
  - -pw
  - -db
  - -df
  - -1f
  - -bf
  - -t
  - -delim
  - -escapeChar
  - -boolStyle
  - -encoding
  - -nullValue
  - -maxErrors

  When **Load Method = ET**, do not use the following load options:
  - remotesource
The following table summarizes the properties of Netezza enterprise stage.

Table 4. Target Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values to select from</th>
<th>Default</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Method</td>
<td>• nzload</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>• ET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Write Mode</td>
<td>• Append</td>
<td>Append</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>• Create</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Replace</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Truncate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update mode</td>
<td>For Update Mode = Update/Update then insert:</td>
<td>True</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>• Unique Key Column</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Key Column</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Updatable Column</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Temporary Work Table</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Drop Work Table</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For Update Mode = Delete/Delete then insert:
• Key Column
• Temporary Work Table
• Drop Work Table

Table 5. Connection properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values to select from</th>
<th>Default</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data source</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Database</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Server</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 5. Connection properties (continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Values to select from</th>
<th>Default</th>
<th>Required?</th>
</tr>
</thead>
</table>
| Temporary Work Table (TWT) in Different Database   | True/False            | False   | • Yes if Drop Work Table = True  
|                                                    |                       |         | • No if Drop Work Table = False |
| TWT Database                                       | N/A                   | N/A     | Yes       |
| User                                               | N/A                   | N/A     | Yes       |

Table 6. Options properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values to select from</th>
<th>Default</th>
<th>Required?</th>
</tr>
</thead>
</table>
| Truncate Column Name                   | • True  
|                                       | • False           | False   | No        |
| Close Command                         | N/A                   | N/A     | No        |
| Drop Unmatched Columns                | True, False          | False   | No        |
| Create Statement                      | N/A                   | N/A     | No        |
| Open Command                          | N/A                   | N/A     | No        |
| Delimiter                             | N/A                   | @       | No        |
| Netezza load options                  | N/A                   | N/A     | No        |

Specify partitioning information for incoming data:
About this task

Use the Partitioning tab to specify how incoming data is partitioned before it is written to the database.

Specify column definitions for incoming data:
About this task

Use the Columns tab to specify column definitions for the table to which the incoming data will be written.

Specify buffering settings for the input link:
About this task

Use the Advanced tab to change default buffering settings for the input link.

Setting attributes for Netezza enterprise Stage
About this task

Select the Stage tab to specify general information about the stage. Setting appropriate attributes for the stage includes:
- Specifying a description of Netezza enterprise stage.
- Specifying how Netezza enterprise stage runs.
- Specifying a character set map for Netezza enterprise stage.

The following sections describe the above tasks in detail.

Specifying a description of Netezza enterprise stage.
This is an optional feature. Use the General tab to enter a description of Netezza enterprise stage.

**Specifying how Netezza enterprise stage runs.**

To specify how Netezza enterprise stage runs, you must use the Advanced tab to set up the following appropriately:

- **Execution Mode**: You can set up Netezza enterprise stage to execute in parallel mode or sequential mode. In parallel mode, the data that you want to load is processed by the available nodes as specified in the Configuration file, and by any node constraints specified using the Advanced tab. By default, Netezza enterprise stage runs in parallel mode.

  In sequential mode, the conductor node processes the data that you want to load.

- **Combinability mode**: Combinability mode is set to Auto by default. Using the Auto mode, Netezza enterprise stage combines various other stages that underlie parallel stages so that those parallel stages run in the same process if they are supported by Netezza enterprise stage.

- **Configuration file**: Use this option to select and set up the appropriate Configuration file. To set up the Configuration file, you must set up the following options.
  - **Node pool and resource constraints**: Select this option for parallel execution of the node pools and resource pools that you have specified.
  - **Node map constraint**: Select this option for parallel execution to the nodes in a defined node map. You can define a node map by directly entering node numbers into the Node map constraint field. Alternatively, click the browse [...] button to select nodes from the Available Nodes... window and select the appropriate nodes from the list.

**Specifying a character set map for Netezza enterprise stage.**

The NLS Map tab appears on the Stage page only if you have NLS (National Language Support) enabled on your system. Use the NLS Map to define a character set map for Netezza enterprise stage. To define the character set map, use any one of the methods below:

- **Method 1**: Enter the desired map name in the Map name field.
- **Method 2**: Select the appropriate map, if available, from the Map name list.
- **Method 3**: Insert the desired value as a job parameter. Click the pointer button on the right side of the Properties page, and then Insert Job Parameters.

A list of available job parameters appears from which you can choose the appropriate parameter. If you wish to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

Using Method 2 overrides the default character set map set for the project or the job that you are currently using.

**Reading data from Netezza Performance Server using Netezza enterprise stage**

**About this task**

To read data from Netezza Performance Server, create a job and then set appropriate properties for Netezza enterprise stage and its output link.
Setting up the attributes of the Output link
About this task
Netezza enterprise stage has an output link that reads data to one table.

When you double-click the Netezza enterprise stage icon from the IBM InfoSphere DataStage palette, a dialog is displayed. This dialog displays two tabs: Stage and Output. The Output tab is enabled by default, and this tab displays the Output page.

Using the Output page, set the attributes that control how data is read from Netezza Performance Server. You perform the following tasks to set up the output link:
• Specifying a description of Netezza enterprise stage
• Setting up the properties of Netezza enterprise stage
• Specifying column definitions for outgoing data
• Specifying buffering settings for the output link

Specifying a description of Netezza enterprise stage:
About this task
Use the General tab on the Output page to specify a description of the output link. This is an optional feature.

Setting up the properties of Netezza enterprise stage:
About this task
On the Output page, the Properties tab is enabled by default. Under the Properties tab, you see a list of properties distributed under 3 categories, Source, Connection, and Options. To edit a property, select the property from the list. You see information about the selected property in the fields next to the list of properties.

Log files
To know about errors that occur during any database operations, and success or failure of these operations, you must verify the log files. For ease of locating the log files, the naming convention of log files for external table and nzload methods are shown:
• Netezza error log files are written to a directory based on the APT_NETEZZA_LOAD_FILES or APT_TMPDIR environment variable that is defined in the order of priority. By default, the error log files are written into .../tmp directory in Netezza Performance Server. The log file names for external table load method are:
  – NZORCH.<timestamp >_<partition number>..<database name>.nzlog and
  – NZORCH.<timestamp >_<partition number>..<database name>.nzbad

Source
Under this category, you must specify values for the following options:
• Read Method
• Table

These properties are described in detail.
• **Read Method**

Specify how to read data. You can use the Table utility, an automatically generated SQL statement or user-defined SQL statement. The default is Table.

• **Table**

Specify the name of the table from which you want to read data.
- Enter the table name in the **Table** field on the right side of the Properties page.
- Insert a value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then **Insert Job Parameters**.

A list of available job parameters appears from which you can select the appropriate parameter. If you want to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

Using the InfoSphere DataStage and QualityStage client, you can also create parameters at the project level for all jobs within the project.

**Connection**

Under this category, you must specify the following values:

• **Data source**: Specify the name of the data source.

• **Database**: Specify the name of the database to which you need to connect. You must enter the same name as the database name specified in the data source. Otherwise, data will not be read.

• **Password**: Specify the password for connecting to the database.

• **Server**: Specify the name of the server that you want to use for extracting data to Netezza Performance Server.

• **User**: Specify the user name for connecting to the data source.

To specify values for the above fields, use any one of the methods:

- Enter the appropriate value in the corresponding field on the right side of the Properties page. For example, when you select **Data source**, you can directly type the required data source name in the **Data source** field that appears on the right side of the page.

- Insert the required value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then **Insert Job Parameters**.

A list of available job parameters appears from which you can select the appropriate parameter. If you want to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

**Options**

Specify the values for the following optional properties:

• **Open Command**

• **Close Command**

• **Delimiter**

• **Open Command**

Specify the appropriate command in single quotes. Before opening the source table, Netezza enterprise stage parses and runs this command on all processing nodes.

To specify a value for **Open Command**, use any one of the methods:
Enter the command in the **Open Command** field on the right side of the Properties page.

Insert the required value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then **Insert Job Parameters**. A list of available job parameters appears from which you can select the appropriate parameter. If you want to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

**Close Command**

Specify the appropriate command in single quotes. Netezza enterprise stage parses and runs this command on all processing nodes after processing the table.

To specify a value for **Close Command**, use any one of the methods:

- Enter the command in the **Close Command** field on the right side of the Properties page.
- Insert the required value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then Insert Job Parameters. A list of available job parameters appears from which you can select the appropriate parameter. If you want to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

If the specified close command fails, an unrecoverable error occurs. This error is logged in the appropriate log file in $DSHOME. However, even when your query fails, the status of the job that you are creating is displayed as OK in the log file. You must visit the log file to find out whether your query was successful.

**Delimiter**

Specify an ASCII character to use as the delimiter for field values or columns. The default delimiter is @. You can use any ASCII character except space (" ") and hyphen ("-").

To specify the delimiter, use any one of the methods:

- Enter the required ASCII character in the Delimiter field on the right side of the Properties page.
- Insert the required value as a job parameter. Click the pointer button on the extreme right side of the Properties page, and then Insert Job Parameters. A list of available job parameters appears from which you can select the appropriate parameter. If you want to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

**Note:** Delimiter is maintained in the graphical user interface to support backward compatibility.

**Specify column definitions for outgoing data:**

**About this task**

Use the **Columns** tab to specify column definitions for the table to which the outgoing data will be read.

**Specify buffering settings for the Output link:**
About this task

Use the Advanced tab to change the default buffering settings for the output link.

Setting attributes for Netezza enterprise Stage:

About this task

Select the Stage tab to specify general information about the stage. Setting appropriate attributes for the stage includes:

- Specifying a description of Netezza enterprise stage.
- Specifying how Netezza enterprise stage runs.
- Specifying a character set map for Netezza enterprise stage.

The following sections describe the above tasks in detail.

Specifying a description of Netezza enterprise stage.

This is an optional feature. Use the General tab to enter a description of Netezza enterprise stage.

Specifying how Netezza enterprise stage runs.

To specify how Netezza enterprise stage runs, you must use the Advanced tab to set up the following appropriately:

- **Execution Mode**: You can set up Netezza enterprise stage to execute in parallel mode or sequential mode. In parallel mode, the data that you want to load is processed by the available nodes as specified in the Configuration file, and by any node constraints specified using the Advanced tab. By default, Netezza enterprise stage runs in parallel mode. In sequential mode, the conductor node processes the data that you want to load.

- **Combinability mode**: Combinability mode is set to Auto by default. Using the Auto mode, Netezza enterprise stage combines various other stages that underlie parallel stages so that those parallel stages run in the same process if they are supported by Netezza enterprise stage.

- **Configuration file**: Use this option to select and set up the appropriate Configuration file. To set up the Configuration file, you must set up the following options.
  - **Node pool and resource constraints**: Select this option for parallel execution of the node pools and resource pools that you have specified.
  - **Node map constraint**: Select this option for parallel execution to the nodes in a defined node map. You can define a node map by directly entering node numbers into the Node map constraint field. Alternatively, click the browse [...] button to select nodes from the Available Nodes... window and select the appropriate nodes from the list.

Specifying a character set map for Netezza enterprise stage.

The NLS Map tab appears on the Stage page only if you have NLS (National Language Support) enabled on your system. Use the NLS Map to define a character set map for Netezza enterprise stage. To define the character set map, use any one of the methods below:

- **Method 1**: Enter the desired map name in the Map name field.
Method 2: Select the appropriate map, if available, from the Map name list.

Method 3: Insert the desired value as a job parameter. Click the pointer button on the right side of the Properties page, and then Insert Job Parameters.

A list of available job parameters appears from which you can choose the appropriate parameter. If you wish to create a new parameter for the job, click [New...] from the list, and create an appropriate environment variable using the Job Properties dialog that appears.

Using Method 2 overrides the default character set map set for the project or the job that you are currently using.

Data types for Netezza enterprise stage

Netezza enterprise stage supports the following data types for read, write, and update/upsert operations.

Read data types

The following table summarizes Netezza read stage data types supported by Netezza Performance Server, ODBC, and equivalent in Orchestrate.

<table>
<thead>
<tr>
<th>Netezza SQL types</th>
<th>ODBC SQL types</th>
<th>Orchestrate Data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>SQLINT4</td>
<td>Int32</td>
</tr>
<tr>
<td>BIGINT</td>
<td>SQLINT8</td>
<td>Int64</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>SQLINT2</td>
<td>Int16</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>SQLNUMERIC (p, s)</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Decimal</td>
<td>SQLDECIMAL</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Real</td>
<td>SQLFLT4</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Double</td>
<td>SQLFLT8</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Char (n)</td>
<td>SQLCHARACTER (n)</td>
<td>string (n)</td>
</tr>
<tr>
<td>Varchar (n)</td>
<td>SQLCHARACTER (n)</td>
<td>string (max=n)</td>
</tr>
<tr>
<td>Nchar (n)</td>
<td>SQLNCHAR</td>
<td>ustring (n)</td>
</tr>
<tr>
<td>Nvarchar (n)</td>
<td>SQLNCHAR</td>
<td>ustring (max=n)</td>
</tr>
<tr>
<td>Date</td>
<td>SQLCHARACTER</td>
<td>Date</td>
</tr>
<tr>
<td>Time</td>
<td>SQLCHARACTER</td>
<td>Time</td>
</tr>
<tr>
<td>Timestamp</td>
<td>SQLCHARACTER</td>
<td>Timestamp (n)</td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>SQLCHAR (1)</td>
<td>char (1)</td>
</tr>
</tbody>
</table>

Write data types

The following table summarizes the Netezza write stage data types supported by Netezza Performance Server, ODBC, and equivalent in Orchestrate.

<table>
<thead>
<tr>
<th>Netezza SQL types</th>
<th>ODBC SQL types</th>
<th>Orchestrate Data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>SQLINT4</td>
<td>Int32</td>
</tr>
<tr>
<td>BIGINT</td>
<td>SQLINT8</td>
<td>Int64</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>SQLINT2</td>
<td>Int16</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>SQLNUMERIC (p, s)</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Decimal</td>
<td>SQLDECIMAL</td>
<td>decimal (p, s)</td>
</tr>
</tbody>
</table>
Table 8. Netezza write stage data types (continued)

<table>
<thead>
<tr>
<th>Netezza SQL types</th>
<th>ODBC SQL types</th>
<th>Orchestrate Data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>SQLFLT4</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Double</td>
<td>SQLFLT8</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Char (n)</td>
<td>SQLCHARACTER (n)</td>
<td>string (n)</td>
</tr>
<tr>
<td>Varchar (n)</td>
<td>SQLCHARACTER (n)</td>
<td>string (max=n)</td>
</tr>
<tr>
<td>Nchar (n)</td>
<td>SQLNCHAR</td>
<td>ustring (n)</td>
</tr>
<tr>
<td>NVarchar (n)</td>
<td>SQLNCHAR</td>
<td>ustring (max=n)</td>
</tr>
<tr>
<td>Date</td>
<td>SQLCHARACTER</td>
<td>Date</td>
</tr>
<tr>
<td>Time</td>
<td>SQLCHARACTER</td>
<td>Time</td>
</tr>
<tr>
<td>Timestamp</td>
<td>SQLCHARACTER</td>
<td>Timestamp (n)</td>
</tr>
</tbody>
</table>

Update/Upsert data types

Netezza Update Stage supports Netezza SQL datatypes listed in the following table.

The following table summaries a one-to-one mapping between Netezza Performance Server, ODBC, and Orchestrate datatypes.

Table 9. Netezza Update/Upsert data types

<table>
<thead>
<tr>
<th>Netezza SQL types</th>
<th>ODBC SQL types</th>
<th>Orchestrate Data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>SQLINT4</td>
<td>Int32</td>
</tr>
<tr>
<td>BIGINT</td>
<td>SQLINT8</td>
<td>Int64</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>SQLINT2</td>
<td>Int16</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>SQLNumeric (p, s)</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Decimal</td>
<td>SQLDECIMAL</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Real</td>
<td>SQLFLT4</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Double</td>
<td>SQLFLT8</td>
<td>decimal (p, s)</td>
</tr>
<tr>
<td>Char (n)</td>
<td>SQLCHARACTER (n)</td>
<td>string (n)</td>
</tr>
<tr>
<td>Varchar (n)</td>
<td>SQLCHARACTER (n)</td>
<td>string (max=n)</td>
</tr>
<tr>
<td>Nchar (n)</td>
<td>SQLNCHAR</td>
<td>ustring (n)</td>
</tr>
<tr>
<td>NVarchar (n)</td>
<td>SQLNCHAR</td>
<td>ustring (max=n)</td>
</tr>
<tr>
<td>Date</td>
<td>SQLCHARACTER</td>
<td>Date</td>
</tr>
<tr>
<td>Time</td>
<td>SQLCHARACTER</td>
<td>Time</td>
</tr>
<tr>
<td>Timestamp</td>
<td>SQLCHARACTER</td>
<td>Timestamp (n)</td>
</tr>
<tr>
<td>Text</td>
<td>SQLText</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Chapter 3. The Netezza connector

With the Netezza connector, IBM InfoSphere Information Server tools can connect to a Netezza database, import database metadata into the InfoSphere Information Server metadata repository, and access data in the database. The connector provides a single way to work with Netezza data when you design a job.

You can use the Netezza connector to:
- Read, write, and look up data in Netezza database tables.
- Create, drop, and truncate tables.
- Perform normal and sparse lookup operations.
- Import the following metadata into a job:
  - Table and column definitions
  - Lists of Netezza databases
  - Tables
  - Views
  - Database aliases
- Work with all Netezza data types with the exception of interval types.
- Work with national language data.

Designing jobs by using the Netezza connector

You can use the IBM Netezza connector to develop jobs that read, write, and load data, and that store components from one job in the repository so that you can reuse them in another job.

Before you begin

Verify that the user name for connecting to the Netezza database has the correct authority and privileges to perform the actions that your job requires.

Procedure
1. Import metadata from a Netezza source.
2. Define a Netezza connector job
3. To set up the Netezza Connector stage to read data from a Netezza table:
   a. Configure the Netezza connector as a source
   b. Set up column definitions
   c. Define usage properties for reading data
   d. Optional: Define how data is partitioned
4. To set up the Netezza Connector stage to write data into a Netezza table:
   a. Configure the Netezza connector as a target
   b. Set up column definitions if they are not already specified for the link.
   c. Define usage properties for writing data
   d. Optional: Define how data is partitioned
5. Look up data by using reference links
6. Compile and run the job
Importing metadata

Metadata import refers to the process of obtaining data about data. Metadata includes information like a list of DSN names, list of databases, list of tables, information about table columns etc. To place table definitions in the repository where they can be used by other projects or components, use the InfoSphere DataStage and QualityStage Designer client to import metadata by using the Netezza connector. When you import metadata from a Netezza data source, a table is created in the metadata repository, and a table definition is created in your project repository tree.

About this task

By using the Netezza connector, you can import the following information:

- Data sources
- Database tables, system tables, and views
- Column definitions for a table, system table or a view

When you import metadata, information about the database columns is collected, including the names of columns, the length of columns, and the data types of columns.

Procedure

1. From the Designer client, open the Connector metadata Import window by selecting **Import > Table Definitions > Start Connector Import Wizard** from the main menu.
2. On the Connector selection page, select the Netezza connector for the import process and click **Next**.
3. On the Connection import page, enter the connection details and click **Next**.
4. On the Data Source Location page, select the host name and database that identifies where you want to store the metadata in the dynamic repository, and click **Next**. If the lists are not populated, click **New location** to start the Shared Metadata Management tool. Use this tool to create host and database objects in the repository that correspond to the data source that you are importing metadata from.
5. On the Filter page, enter or select the required filter information for the import and click **Next**. This page collects information that is specific to the type of connector that you are using for the import process.
6. On the Selection page specify the items that you want to import and click **Next**.
7. On the Confirm import page to confirm the import details, click **Import**.
8. Browse the repository tree, and select the location in the project repository for the table definition that you are creating. Click **OK**.

Netezza identifiers

In Netezza connector, identifiers represent table names that are entered in various table name properties, or are referenced in the manually entered SQL statements. Identifiers also represent table columns.

Netezza supports the SQL 1999 definition for naming identifiers, and they can be up to 128 bytes in length. There are two types of identifiers:

- Regular
- Delimited
The Netezza connector supports both regular and delimited identifiers in both Table name property and user-defined statements.

**Regular identifiers**

Regular identifiers are case insensitive and are stored in the database in upper-case. Regular identifiers must begin with a letter; they cannot begin with a digit, underscore, or dollar sign. Regular identifiers also cannot be a SQL reserved word. The encoding in the Netezza catalog is in UTF-8; the encoding for any display will depend on the client. Regular identifiers can contain only the following:

- letters (in any alphabet, not just the Latin alphabet)
- syllables (as in the Japanese Hiragana syllabary)
- ideographs
- decimal digits
- underscores
- dollar sign

**Delimited identifiers**

Delimited identifiers are enclosed in double-quotation marks and have special considerations. They are case-sensitive. The Netezza system does not convert delimited identifiers to the default system case, nor does it save the enclosing double-quotation marks in the database. Within the double quotation marks, a delimited identifier can include the same letters, syllables, ideographs, decimal digits, and underscores as a regular identifier, but it can also include spaces, special characters such as hyphens and percent signs, and SQL reserved keywords. A delimited identifier can begin with any letters, digits, or symbols.

**Referencing Database Objects**

Netezza supports two-level and three-level naming.

Two-level naming is used to reference an object in the same database. Example, `schema.object-name`.

Three-level naming is used to reference an object in another database residing on the same physical Netezza server. Example, `database-name.schema.object-name`.

A short-hand three-level form is also supported. Example, `database-name..object-name`. In this case the schema name is implied and is set to the current schema name.

*Note:* Cross database access is only supported for reading. Modifying a table in a different database is not supported by Netezza.

**Supported character encoding**

InfoSphere DataStage National Language Support (NLS) makes it possible for you to process data in international languages using Unicode character sets. Verify that the operating system of the target computer uses a supported character encoding.

Netezza server supports the following two character encoding:

- Latin-9 – CHAR and VARCHAR
UTF-8 – NCHAR and NVARCHAR

This cannot be configured. The same encodings are used when the data is sent or received through a pipe or file.

**Defining a Netezza connector job**

Use the InfoSphere DataStage and QualityStage Designer client to define a job by using the Netezza connector.

**Procedure**

1. From the Designer client, select **File > New** from the menu.
2. In the New window, select the **Parallel Job** or **Server Job** icon, and click **OK**.
3. On the left side of the Designer client in the Palette menu, select the **Database** category.
4. Locate **Netezza Connector** in the list of available databases, and click the down arrow to view the available stages.
5. Drag the **Netezza Connector** stage icon to the job design canvas.
6. Enter or modify the following attributes:
   - **Name of the Netezza Connector stage or link**: Modify the default name of the connector or the link. You can enter up to 255 characters. Alternatively, you can modify the name of the stage or link in the job design canvas.
   - **Description**: Enter an optional description of the stage or link.
7. Click **Save**.

**What to do next**

Define properties to use Netezza connector as a source.

**Reading data**

To read data from a Netezza table by using the Netezza connector, you need to configure the Netezza connector to process data as a source. The Netezza connector runs an SQL statement and returns the results as a set of zero or more rows. As a source, the connector extracts or reads data from an external Netezza data source.

The following figure shows an example of using the Netezza connector to read data. In this case, the Netezza connector **Netezza_Connector_1** reads data from the Netezza server and then the sequential file stage writes it to the file **Sequential_File_3**. When you configure the Netezza connector to read data, you create only one output link **DSoutputLink**, which is shown in the figure below transferring rows from **Netezza_Connector_1** to **Sequential_File_3**.
Configuring the Netezza connector as a source:

By configuring the Netezza connector to process data as a source, you can use the Netezza Connector stage to read data.

**Procedure**
1. On the job design canvas, double-click the Netezza Connector stage icon.
2. Click the **Output** tab, then select the output link that you want to edit. By editing the output link you are setting up the Netezza Connector stage to be the source.
3. Select the **Properties** tab.
4. In the **Connection** section, specify the **Data source**, **Database**, **User Name**, and **Password** that you want to use to make the connection.
5. Click **Save** to save the connection settings that you specified.

Setting up column definitions:

You set up column definitions for read operations and write operations in a similar way. You can also customize the columns grid, save column definitions for later use, and load predefined column definitions from the repository.

**Procedure**
1. On the job design canvas, double-click the Netezza connector icon.
2. In the stage editor, select either the output tab or the input tab.
3. On the **Columns** tab, modify the columns grid to specify the metadata that you want to define.
   a. Right-click within the grid, and select **Properties** from the menu.
   b. In the Grid properties window, select the properties that you want to display and the order that you want them to be displayed. Then, click **OK**.
4. Enter column definitions for the table by using one of the following methods:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Method 1        | 1. In the **Column name** column, double-click inside the appropriate cell and type a column name.  
2. For each cell in the row, double-click inside the cell and select the options that you want.  
3. In the **Description** column, double-click inside the appropriate cell and type a description. |
| Method 2        | 1. Right-click within the grid, and select **Edit row** from the menu.  
2. In the Edit column metadata window, enter the column metadata.                                                                                                     |

5. To share metadata between several columns, select the columns that you want to share metadata.
   a. Right-click and select **Propagate values**.
   b. In the Propagate column values window, select the properties that you want the selected columns to share.

6. To save the column definitions as a table definition in the repository, click **Save**.
   a. Enter the appropriate information in the Save Table Definition window, and then click **OK**.
   b. In the Save Table Definition As window, select the folder where you want to save the table definition, and then click **Save**.

7. To load column definitions from the repository, click **Load**.
   a. In the Table Definitions window, select the table definition that you want to load, and then click **OK**.
   b. In the Select Columns window, use the arrow buttons to move columns from the **Available columns** list to the **Selected columns** list. Click **OK**.

**Defining properties for reading data:**

You must configure how the Netezza connector operates in a job when reading data by defining the usage properties.

**Before you begin**

You must configure a database connection (as a source) for the Netezza connector.

**Procedure**

1. On the job design canvas, double-click the **Netezza connector** icon.
2. Click the **Output** tab, then select the output link that you want to edit.
3. Click the **Properties** tab.
4. On the **Properties** tab in the **Usage** section, specify how the connector operates in a job.
   a. Specify whether you want SQL statements generated at run time in the **Generate SQL** field.
   b. In the **Table** field, specify the table that you want to read.
   c. In the **Enable quoted identifiers** field, specify **Yes** to retain the case of all of the object names in DDL and DML statements. The default is **No**.
d. In the Before or After SQL field, specify whether an SQL statement runs before or after data processing.

5. Click OK to save your changes.

Partitioning data for parallel reads for the Netezza connector:

You must configure the Netezza connector to perform parallel reads; by default the connector runs sequentially. A parallel read is when the data is divided into subsets of data, and then the data is concurrently read by different processing nodes. The Netezza connector supports modulus partitioning. With modulus partitioning the rows are distributed between the processing nodes by adding a modulus expression against the special Netezza column `datasliceid` to the WHERE clause. For more information about the partition configuration and logical nodes, see the Parallel Job Developer’s Guide.

About this task

Procedure

1. On the job design canvas, double-click the Netezza connector icon.
2. Select the reference link (for normal lookups only) from the Link drop-down list and then select the Lookup Type as Normal, or, select the output link that you want to configure for the parallel reads.
3. Set Enable partitioned reads to Yes.
4. Click OK.

Results

When this job is run, the WHERE clause in your SELECT statement is modified to return a subset of rows that are read by each processing node.

Example

If there are four processing nodes the SELECT statements in each processing node, it displays as follows:

```
SELECT col1, col2 FROM table WHERE mod(datasliceid,4)=0
SELECT col1, col2 FROM table WHERE mod(datasliceid,4)=1
SELECT col1, col2 FROM table WHERE mod(datasliceid,4)=2
SELECT col1, col2 FROM table WHERE mod(datasliceid,4)=3
```

If you use a user-defined SQL and the connector reads in parallel, it is your responsibility to ensure appropriate partitioning. To aid in that process the connector defines two place holders that are replaced with actual values at run time:

<table>
<thead>
<tr>
<th>Place holder</th>
<th>Description</th>
<th>Value when running sequentially</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>[[node-count]]</code></td>
<td>The total number of processing nodes. This place holder represents the level of parallelism for the Netezza connector stage and is equal to the number of processing nodes.</td>
<td>1</td>
</tr>
<tr>
<td>Place holder</td>
<td>Description</td>
<td>Value when running sequentially</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>[[node-number]]</td>
<td>The current processing node zero-based index. For example, if there are 4 processing nodes, the processing node indexes are 0,1,2,3.</td>
<td>0</td>
</tr>
</tbody>
</table>

You can use the place holders to create partitioning clauses in your SQL statements. For example:

```sql
SELECT * FROM table WHERE mod(datasliceid,[[node-count]])=[[node-number]]
```

`[[node-count]]` is replaced with the total number of processing nodes and `[[node-number]]` is replaced with the current processing node zero-based index.

**Note:** The column `datasliceid` is a special Netezza column specific to each table and is not automatically included in views and if needed must be explicitly added to the select list of the view query.

### Configuring normal lookup operations:

Data that is read by a database stage can serve as reference data to a Lookup stage. By default, this reference data is loaded into memory like any other reference link. When you perform a normal lookup operation, the Netezza connector retrieves all of the records and allows the Lookup stage to process the records.

**Before you begin**

You must create a lookup operation job first. You also must define your columns in the input stage for the Lookup stage and the output stage for the Lookup stage.

**Procedure**

1. Double-click the Netezza Connector stage to open the link properties window.
2. From **Lookup Type** drop-down list, select **Normal**.
3. Click the **Columns** tab, and define the columns that you want to use from the database to which the connector is connected.
4. Configure the properties on the **Properties** tab.
   a. Define and test your connection properties in the **Connection** section.
   b. In the **Usage** section, you can specify if you want auto-generated SQL statements or user-defined SQL statements. Specify **Generate SQL** as **Yes** or **No**.
      - If you specify **Generate SQL** as **Yes**, specify the **Table name** and then select the **Key** columns in the **Columns** tab in the lookup stage. To specify the **Key** columns drag the required columns from the primary link to the reference link. This configuration means that many records are retrieved as opposed to the records that are retrieved in a sparse lookup operation.
      - If you specify **Generate SQL** as **No**, select the **Key** columns in the **Columns** tab in the lookup stage. To specify the **Key** columns drag the required columns from the primary link to the reference link. Specify the **Select statement** property. Type your SELECT statement in the **Select statement** property using the following format: `select * from table_name`. This configuration means that many records are retrieved as opposed to the records that are retrieved in a sparse lookup operation.
c. Optional: Configure any other properties on the **Properties** tab.

5. Click **OK** to save the changes.

6. To map the input links to the output link, double-click the Lookup stage to open the stage editor.
   a. Drag or copy the columns from your input link to your output link to add the columns to the output link.
   b. Define any conditions for a lookup failure by clicking the **Constraint** icon in the menu.
   c. Select the appropriate value for the **Lookup Failure** column and click **OK**. If you select **Reject**, you must have a reject link and target stage in your job configuration to capture these records.

7. Click **OK**.

8. Save, compile, and run the job.

**Configuring sparse lookup operations:**

Data that is read by a database stage can serve as reference data to a Lookup stage. By default, this reference data is loaded into memory like any other reference link. If you configure the Netezza connector to directly connect as the reference link to a Lookup stage, you can configure the **Lookup Type** property of the Netezza connector to Sparse and send individual SQL statements to the database for each incoming row in the lookup table.

**Before you begin**

You must create a lookup operation job.

**About this task**

If the number of input rows to a stage is significantly smaller than the number of reference rows (1:100 or more) in a database table, you can configure the Netezza connector to perform a sparse lookup operation and send individual SQL statements to the database for each incoming row in the lookup table. In the lookup operation job, the connector receives the records from the input stage, and then the connector performs the lookup operation directly on the external resource. The connector then generates the output records.

You can use the sparse lookup method only in parallel jobs.

**Procedure**

1. Double-click on the Netezza Connector stage to open the link properties window.
2. Select **Lookup Type** drop-down list, select **Sparse**.
3. Click the **Columns** tab, and define the columns that you want to use from the database to which the connector is connected.
4. Configure the properties on the **Properties** tab.
   a. Define and test your connection properties in the **Connection** section.
   b. In the **Usage** section, you can specify if you want auto-generated SQL statements or user-defined SQL statements. Specify **Generate SQL** as **Yes** or **No**.
      • If you specify **Generate SQL** as **Yes**, specify the **Table name** and the **Key** columns details in the **Columns** tab.
If you specify Generate SQL as No, specify the Select statement property. In the select part of the SELECT statement list the columns you want to return to the job. Normally this list would match the columns in the Columns tab. If the list does not match the columns, you need to select the appropriate Schema reconciliation properties in the Properties tab of Input page to achieve the desired behavior. The following syntax is an example of the first part of the SELECT statement: select Field001,Field002,Field003.

c. Specify Table name in the Properties tab and then specify a WHERE clause to perform the lookup. Key columns that follow the WHERE clause must have the word ORCHESTRATE and a period added to the beginning of the column name. ORCHESTRATE can be capitalized or lowercase letters, such as: ORCHESTRATE.Field001. The following SELECT statement is an example of the correct syntax of the WHERE clause: select Field001,Field002,Field003 from MY_TABLE where ORCHESTRATE.Field001 = Field001. The column names following the keyword ORCHESTRATE should match the column names in the Columns tab. If the list does not match the columns, you need to select the appropriate Schema reconciliation properties in the Properties tab of Input page to achieve the desired behavior.

d. Optional: Configure any other properties on the Properties tab.

5. Click OK to save the changes.

6. To map the input links to the output link, double-click the Lookup stage to open the stage editor.

   a. Drag or copy the columns from your input link to your output link to add the columns to the output link.

   b. Define any conditions for a lookup failure by clicking the Constraint icon in the menu.

   c. Select the appropriate value for the Lookup Failure column and click OK. If you select Reject, you must have a reject link and target stage in your job configuration to capture these records.

7. Click OK.

8. Save, compile, and run the job.

Writing data

To write data to a Netezza table by using the Netezza connector, you configure the Netezza connector to process data as a target. The Netezza connector runs an SQL statement and connects to the external Netezza data source and inserts, updates, or deletes data as required.

The following figure shows an example of using the Netezza connector to write data. In this case, the sequential file stage reads data from the file Sequential_File_1 and then the Netezza connector stage inserts, updates, or deletes data into Netezza database as required.
Configuring the Netezza connector as a target:

By configuring the Netezza connector to process data as a target, you can use the Netezza Connector stage to write data.

**Procedure**

1. On the job design canvas, double-click the **Netezza Connector** stage icon.
2. Click the **Input** tab, then select the input link that you want to edit. By editing the input link you are configuring the Netezza Connector stage to be the target.
3. Select the **Properties** tab.
4. In the **Connection** section, specify the **Data source, Database, User Name, and Password** that you want to use to make the connection.
5. Specify **Use separate connection for TWT** to indicate if you want to use a separate connection and include the Temporary Work Table (TWT) and External tables (ETs) in a separate database.
6. Click **Save** to save the connection settings that you specified.

**Setting up column definitions:**

You set up column definitions for read operations and write operations in a similar way. You can also customize the columns grid, save column definitions for later use, and load predefined column definitions from the repository.

**Procedure**

1. On the job design canvas, double-click the **Netezza connector** icon.
2. In the stage editor, select either the output tab or the input tab.
3. On the **Columns** tab, modify the columns grid to specify the metadata that you want to define.
   a. Right-click within the grid, and select **Properties** from the menu.
   b. In the Grid properties window, select the properties that you want to display and the order that you want them to be displayed. Then, click **OK**.
4. Enter column definitions for the table by using one of the following methods:
Option | Description
--- | ---
Method 1 | 1. In the **Column name** column, double-click inside the appropriate cell and type a column name.
2. For each cell in the row, double-click inside the cell and select the options that you want.
3. In the **Description** column, double-click inside the appropriate cell and type a description.

Method 2 | 1. Right-click within the grid, and select **Edit row** from the menu.
2. In the Edit column metadata window, enter the column metadata.

5. To share metadata between several columns, select the columns that you want to share metadata.
   a. Right-click and select **Propagate values**.
   b. In the Propagate column values window, select the properties that you want the selected columns to share.

6. To save the column definitions as a table definition in the repository, click **Save**.
   a. Enter the appropriate information in the Save Table Definition window, and then click **OK**.
   b. In the Save Table Definition As window, select the folder where you want to save the table definition, and then click **Save**.

7. To load column definitions from the repository, click **Load**.
   a. In the Table Definitions window, select the table definition that you want to load, and then click **OK**.
   b. In the Select Columns window, use the arrow buttons to move columns from the **Available columns** list to the **Selected columns** list. Click **OK**.

**Configuring duplicate row handling:**

The Netezza connector handles duplicates by filtering duplicates in the input data and also by inserting only rows that do not exist in the target table. You can configure the connector to check for duplicate rows. The connector uses additional SQL logic to detect duplicate rows.

**About this task**

The connector groups rows by the key values and orders them based on the value of the ROWID column. The row added recently has a higher ROWID. After the rows are ranked within every key combination, the connector selects one of the combination and filters out the rest. When **Use unique key column** is set to **Yes**, the connector uses the specified column in place of ROWID.

**Procedure**

1. Double-click the connector stage icon to open the connector properties.
2. Select the **Input** tab.
3. Select the **Properties** tab.
4. Specify an option for **Write mode**.
5. Set the **Check duplicate rows** option to **Yes**.

6. Set the **Duplicate row action** option to **Filter**. The connector keeps only one row for each key combination and filters out the rest. When the option is set to **Fail**, the job fails if any duplicates are detected.

Results

The following table summarizes the behavior of connector for different write modes, when duplicate row handling is enabled:

<table>
<thead>
<tr>
<th>Write mode</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>Selects the first row for each key combination and ignores the rest. If the key is not already in the target table, the selected row is inserted. Else, no action is taken.</td>
</tr>
<tr>
<td>Update</td>
<td>Selects the last row for each key combination and ignores the rest. If the matching row exists in the target table, the selected row is used to update the target table. Else, no action is taken.</td>
</tr>
<tr>
<td>Delete</td>
<td>No change in behavior. All matching rows are deleted.</td>
</tr>
<tr>
<td>Update then insert</td>
<td><strong>Update</strong>: Selects the last row for each key combination and ignores the rest. If the matching row exists in the target table, the selected row is used to update the target table. Else, no action is taken. <strong>Insert</strong>: Selects the last row for each key combination and ignores the rest. If the key is not already in the target table, the selected row is inserted. Else, no action is taken.</td>
</tr>
<tr>
<td>Delete then insert</td>
<td><strong>Delete</strong>: No change in behavior. All matching rows are deleted. <strong>Insert</strong>: Selects the last row for each key combination and ignores the rest. Inserts the selected row into the target table.</td>
</tr>
<tr>
<td>Action column</td>
<td>Breaks the input rows into groups for execution in different SQL statements based on the value of the action column and other factors. It ensures that the result is identical to the result you get if each row was applied one at a time in the original incoming order.</td>
</tr>
</tbody>
</table>

Configuring record ordering:

By default, the Netezza connector inserts records into the target table without any explicit ordering. You can enable the record ordering property to specify one or more order columns.

About this task

The record ordering option is available for Insert, Update then insert, Delete then insert, and Action column write modes. When you enable the record handling
option, the records are sorted before they are inserted from TWT into the target table.

**Procedure**

1. Double-click the connector stage icon to open the connector properties.
2. Select the **Input** tab.
3. Select the **Properties** tab.
4. Specify an option for **Write mode**.
5. If you selected the **Insert** write mode, then set the **Direct insert** option to **No**.
6. Set the **Enable record ordering** option to **Yes**.
7. Select the **Column name** property, and then click the **Select column** button to specify the name of the first order key column. The first key property is automatically enabled and required.
8. Select the column of your choice from the list.

**Note:** You must define at least one column on the link for this option to work.

9. Optional: Right-click **Key[1]**, select **Add Property Value**, and then add additional key columns.
10. Right-click **Key[n]** property (where n is the index of the key you want to delete) and select **Remove Property Value**, to delete a key.

**Defining properties for writing data:**

You must configure how the Netezza connector operates in a job when writing data.

**Before you begin**

You must configure a database connection (as a target) for the Netezza connector.

**Procedure**

1. On the job design canvas, double-click the **Netezza connector** icon.
2. Click the **Input** tab, then select the input link that you want to edit.
3. Click the **Properties** tab.
4. On the **Properties** tab in the **Usage** section, specify how the connector operates in a job.
   a. Specify the **Write mode**.
   b. In the **Table name** field, specify the name of the destination table that is used in the SQL statements that are meant for writing data. This property is mandatory. You can create the table at runtime using the **Create** or **Replace** table actions. The table name is used to generate Data Definition Language (DDL) statements. You must specify **Generate Create/Drop/Truncate statement at runtime** property to **Yes** to generate DDL statements at runtime. You must specify **Generate Create/Drop/Truncate statement at runtime** property to **No** to use user-defined DDL statements.
   c. Specify whether you want SQL statements generated at run time in the **Generate SQL** field.
   d. In the **Enable quoted identifiers** field, specify **Yes** to retain the case of all of the object names in DDL and DML statements. The default is **No**.
   e. In the **SQL** field, specify the appropriate SQL statements.
f. In the **Table action** field, specify how you want tables to be created, or how you want rows to be edited or inserted in an existing destination table.

g. In the **Before or After SQL** field, specify whether an SQL statement runs before or after data processing.

5. Click **OK**.

*Partitioning data for a write operation:*

You can specify how the incoming data is partitioned or collected before the data is written to the Netezza database. You can also specify that the data should be sorted before being written.

**Before you begin**

The availability of partition or collection type depends on whether you design your Netezza connector job to run in parallel or sequential mode.

- The **Partition type** list is available if you set the Netezza Connector stage to run in parallel mode. If you select a method from the list, the method overrides any current partitioning method.
- The **Collection type** list is available if you set the Netezza Connector stage to run in sequential mode, and the preceding stage is set to run in parallel mode.

**Procedure**

1. On the job design canvas, double-click the **Netezza connector** icon.
2. In the top left corner of the stage editor, select the input link that you want to edit.
3. Click the **Partitioning** tab.
4. Select a partition type from the **Partition type/Collection type** list. Select the type of data partitioning or collection method to the data that you want to write. Click the **Properties** icon next to the **Partition type** list to specify the required database, instance, and table.
5. Click **OK**.

*Upsert operations using temporary work table in a separate database:*

The Netezza connector supports the ability to have the temporary work table (TWT) in another database. To perform any of the upsert operations, the Netezza connector uses a temporary work table (TWT) to collect the input data before executing the SQL statement.

If you put the temporary work table in a separate database it helps in easier maintenance and space management. There is a limitation in Netezza that prevents modifying tables in another database, so the connector uses a separate connection for writing to the temporary work table when using external tables. This feature is activated when you specify the input connection property **Use separate connection for TWT** to **Yes** and configure the parameters. When enabled, the connector establishes another connection and uses it to write data to the TWT.

**Upsert operations**

To perform any of the upsert operations (update, delete, insert then update, update then insert, or delete then insert) the connector uses a temporary work table (TWT) to collect the input data before executing the SQL statement. Using TWT guaranties a predictable result of the upsert statement especially when there are
multiple processing nodes feeding data. The following figure illustrates an upsert operation:

The connector writes the input data (keys and any update columns) to the pipe in one thread and executes an INSERT statement (into TWT) in the other, for example:

\[
\text{INSERT INTO TemporaryWorkTable SELECT * FROM ExternalTable}
\]

The connector creates one named pipe and one external table per processing node. The number of processing nodes is dictated by the number of logical nodes defined in the APT configuration file. The connector then opens the named pipe and starts the execution of the INSERT statement in another thread which blocks until all the data is processed. The connector creates a single TWT regardless of the number of processing nodes. The connector then writes the input data to the pipe. When there is no more input data, the connector closes the pipe which causes the INSERT statement to complete and the connector commits the transaction. Once all processing nodes complete inserting into the TWT the connector executes the upsert statement from the conductor process.

**Looking up data by using reference links**

You can use the IBM Netezza connector to look up data directly from a Netezza table by using a reference link to link the Netezza Connector stage to a Lookup stage. The Lookup stage is a processing stage. It is used to perform lookup operations on a data set read into memory from any other Parallel job stage that can output data. You can specify a condition on each of the reference links, such that the stage will only perform a lookup on that reference link if the condition is satisfied.

**About this task**

A reference link represents a table lookup operation. You can use a reference link as an input link to a Lookup stage and as an output link from other types of stages, such as the Netezza Connector stage.

**Procedure**

1. On the job design canvas, drag a **Netezza connector** icon and a **Lookup stage** icon to the job design canvas. (The **Lookup stage** is located in the **Processing** category of the **Palette** menu.)
2. Join the stages by dragging a link from the Netezza Connector stage to the Lookup stage.
3. Right-click the link, and select Convert to Reference from the menu. The line changes to a dashed line to show that the link is a reference link.
4. Open the Netezza Connector stage editor by double-clicking the Netezza connector icon.
5. In the stage editor, define the database connection information for the stage, and then define the read operation for the reference link. When you are finished, click OK.
6. Select the lookup type as Normal or Sparse.
7. Open the Lookup stage editor by double-clicking the Lookup stage icon.
8. Define the lookup operation for the Lookup stage. The left pane of the editor shows the input links, and the right pane shows the output links. The metadata for these links is displayed below these panes. For each record of the source data set from the primary link, the Lookup stage performs a table lookup on each of the lookup tables attached by reference links. The table lookup is based on the values of a set of lookup key columns, one set for each table.
9. Click OK.

Generating SQL statements in the connector at design time
You can configure the connector to generate SQL statements at design time in their statement properties.

Before you begin
Create a job that includes a connector as a source or target.

About this task
You can generate the SQL statement text only for those statement properties that have the Generate SQL statement option in the Build list.

Note: Under some circumstances, the connector requires a connection to generate SQL statements. When a user name and password are not supplied and a connection is required, a connection is made by using the user who is running the ASB Agent service.

Procedure
1. Double-click the connector on the job canvas to open the stage editor.
2. In the navigator, click the output or input link, depending on the type of job that you create.
3. Set Generate SQL at runtime to No.
4. In the Table name property, type the name of the table for the SQL statement.
5. For jobs in target context (input links), select the type of statement you want to generate in the Write mode property.
6. On the Columns page, define the columns to use in the SQL statement.
7. Click the Properties tab.
8. Click the Build button that is associated with the statement property, and select Generate SQL statement from the list.
**Note:** The *Generate SQL statement* option will only be available for statements which that connector supports generating at design time. In some cases a connector may only support generating the SQL at runtime during job execution.

9. Click **OK** to save the job.

**Validating SQL statements in the connector at design time**

After you generate or write a SQL statement, you can validate the statement during job design.

**About this task**

You can validate the SQL statement text only for those statement properties that have the **Validate SQL** option in the Build list.

**Note:** Under some circumstances, the connector requires a connection to validate SQL statements. When a user name and password are not supplied and a connection is required, a connection is made by using the user who is running the ASB Agent service.

**Procedure**

1. Save the job.

2. Click the **Build** button that is associated with the statement property, and select **Validate SQL**. The **Validate SQL** option is enabled only if the statement property contains a value and this option will only be available for statements which the target RDBMS supports validating.

**Results**

The connector validates the SQL statement by preparing the statement with the RDBMS it supports. If the SQL contains error, an error message is shown.

**Data type conversions**

IBM InfoSphere DataStage supports a set of SQL data types that are different from Netezza SQL data types.

When you import metadata through the Netezza connector or read data by using the Netezza connector, the Netezza Connector stage converts the Netezza data types to InfoSphere DataStage data types. Conversely, when you write data to a target Netezza table through the Netezza connector, InfoSphere DataStage data types are converted to Netezza data types.

**Data type conversions from Netezza to DataStage:**

When reading data, the Netezza Connector stage converts Netezza data types to InfoSphere DataStage data types.

Likewise, after metadata is imported through the Netezza connector, the Netezza data types are converted to IBM InfoSphere DataStage data types. The following table shows the mapping rules between Netezza data types and InfoSphere DataStage data types.
Table 10. Netezza data types and their equivalent InfoSphere DataStage data types

<table>
<thead>
<tr>
<th>Netezza data types</th>
<th>InfoSphere DataStage data types (SQL types)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYTEINT</td>
<td>TinyInt</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>SmallInt</td>
</tr>
<tr>
<td>INT</td>
<td>Integer</td>
</tr>
<tr>
<td>BIGINT</td>
<td>BigInt</td>
</tr>
<tr>
<td>NUMERIC(p, s)</td>
<td>Numeric</td>
</tr>
<tr>
<td>FLOAT(p)</td>
<td>Float</td>
</tr>
<tr>
<td>REAL</td>
<td>Real</td>
</tr>
<tr>
<td>DOUBLE PRECISION</td>
<td>Double</td>
</tr>
<tr>
<td>CHAR(n)</td>
<td>Char</td>
</tr>
<tr>
<td>VARCHAR(n)</td>
<td>VarChar</td>
</tr>
<tr>
<td>NCHAR(n)</td>
<td>NChar</td>
</tr>
<tr>
<td>NVARCHAR(n)</td>
<td>NVarchar</td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>Bit</td>
</tr>
<tr>
<td>DATE</td>
<td>Date</td>
</tr>
<tr>
<td>TIME</td>
<td>Time</td>
</tr>
<tr>
<td>TIME WITH TIME ZONE</td>
<td>VarChar</td>
</tr>
</tbody>
</table>

Note: The value of time with time zone will be returned without the time zone information.

| TIMESTAMP                | TimeStamp                                   |
| INTERVAL                 | VarChar                                     |

Note: You cannot load the interval data type from an external table.

| ROWID                    | BigInt                                      |
| TRANSACTION ID           | BigInt                                      |
| DATASLICE                | Integer                                     |

The XML data type can be imported as a LOB by selecting the XML column as LOB option during metadata import. XML columns can be represented as any of the string, ustring, or LOB data types.

Data type conversions from DataStage to Netezza:

When writing data, the Netezza Connector stage converts InfoSphere DataStage data types to Netezza data types.

In some cases, an exact conversion from an InfoSphere DataStage data type to a Netezza data type does not exist.

The following table shows the mapping rules between InfoSphere DataStage data types and Netezza data types.

Table 11. InfoSphere DataStage data types and their equivalent Netezza data types

<table>
<thead>
<tr>
<th>InfoSphere DataStage data types (SQL type)</th>
<th>Netezza data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>BOOLEAN or BOOL</td>
</tr>
</tbody>
</table>
Table 11. InfoSphere DataStage data types and their equivalent Netezza data types (continued)

<table>
<thead>
<tr>
<th>InfoSphere DataStage data types (SQL type)</th>
<th>Netezza data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char</td>
<td>CHAR(64000), CHAR(n), NCHAR(16000), NCHAR(n)</td>
</tr>
<tr>
<td>VarChar</td>
<td>VARCHAR(64000), VARCHAR(n), NVARCHAR(16000), NVARCHAR(n)</td>
</tr>
<tr>
<td>LongVarChar</td>
<td>VARCHAR(64000), VARCHAR(n), NVARCHAR(16000), NVARCHAR(n)</td>
</tr>
<tr>
<td>NChar</td>
<td>NCHAR(16000), NCHAR(n)</td>
</tr>
<tr>
<td>NVarChar</td>
<td>NVARCHAR(16000), NVARCHAR(n)</td>
</tr>
<tr>
<td>LongNVarChar</td>
<td>NVARCHAR(16000), NVARCHAR(n)</td>
</tr>
<tr>
<td>Binary</td>
<td>Unsupported</td>
</tr>
<tr>
<td>VarBinary</td>
<td>Unsupported</td>
</tr>
<tr>
<td>LongVarBinary</td>
<td>Unsupported</td>
</tr>
<tr>
<td>Decimal</td>
<td>NUMERIC(p), NUMERIC(p, s)</td>
</tr>
<tr>
<td>Double</td>
<td>DOUBLE PRECISION or FLOAT(15)</td>
</tr>
<tr>
<td>Float</td>
<td>REAL or FLOAT(6), FLOAT(p)</td>
</tr>
<tr>
<td>Real</td>
<td>REAL or FLOAT(6)</td>
</tr>
<tr>
<td>TinyInt</td>
<td>BYTEINT</td>
</tr>
<tr>
<td>SmallInt</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>Integer</td>
<td>INT</td>
</tr>
<tr>
<td>BigInt</td>
<td>BIGINT</td>
</tr>
<tr>
<td>Numeric</td>
<td>NUMBER(38,0), NUMBER(p, 0), NUMBER(38, s), NUMBER(p, s)</td>
</tr>
<tr>
<td>Date</td>
<td>DATE</td>
</tr>
<tr>
<td>Time</td>
<td>TIME (always includes microseconds)</td>
</tr>
<tr>
<td>Timestamp</td>
<td>TIMESTAMP (always includes microseconds)</td>
</tr>
<tr>
<td>Unknown</td>
<td>VARCHAR(64000)</td>
</tr>
</tbody>
</table>

Compiling and running Netezza connector jobs

You compile Netezza connector jobs into executable scripts that you can schedule and run.

Procedure

1. In the InfoSphere DataStage and QualityStage Designer client, open the job that you want to compile.
2. Click the Compile button.
3. If the Compilation Status area shows errors, edit the job to resolve the errors. After resolving the errors, click the Re-compile button.
4. When the job compiles successfully, click the Run button, and specify the job run options:
   a. Enter the job parameters as required.
   b. Click the Validate button to verify that the job will run successfully without actually extracting, converting, or writing data.
   c. Click the Run button to extract, convert, or write data.
5. To view the results of validating or running a job:
   a. In the Designer client, select **Tools > Run Director** to open the Director client.
   b. In the Status column, verify that the job was validated or completed successfully.
   c. If the job or validation fails, select **View > Log** to identify any runtime problems.

6. If the job has runtime problems, fix the problems, recompile, validate (optional), and run the job until it completes successfully.

---

**Troubleshooting**

When using the Netezza connector, you might encounter errors that can be fixed by troubleshooting and adjusting values for properties or configuration.

**Connection errors**

You might encounter the following connection errors:

<table>
<thead>
<tr>
<th>Error type</th>
<th>Error details</th>
<th>Troubleshooting details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODBC Driver Cannot Be Loaded</td>
<td>When running a Netezza connector DataStage job, or running a Metadata Import Wizard, or performing View data, you might encounter the following error: Connection to database testdb with user testuser using DSN NZSQL failed. Reason: [SQLCODE=1M003] [Native=0] [DataDirect] [ODBC 1 lib] Specified driver could not be loaded.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reason for the error might be as follows:</td>
<td>For the error in a Windows 64-bit system:</td>
</tr>
<tr>
<td></td>
<td>• On Windows 64-bit – A 64-bit version of the Netezza driver is installed. This does not work because Information Server is a 32-bit application on Windows.</td>
<td>1. Make sure that both 32-bit and 64-bit drivers are installed.</td>
</tr>
<tr>
<td></td>
<td>• On Unix, the library search path is not set correctly.</td>
<td>2. Make sure that you define your DSNs using the 32-bit version of Data source (ODBC). The 32-bit version of the Odbcad32.exe file is located in the %systemdrive%\Windows\ SysWoW64 folder. See the following article for more details of DSN-related issues on Windows 64-bit: <a href="http://support.microsoft.com/kb/942976">http://support.microsoft.com/kb/942976</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>For the error in a Unix system, include the Netezza ODBC driver directory (&lt;nz-odbc-home&gt;/lib64) in the path search environment variable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On an AIX system, if you get the error even after the library search path is set properly, you need to go to the Netezza ODBC driver directory (&lt;nz-odbc-home&gt;/lib64) and create the following symbolic link (depending on your setup you might need root privileges): cd &lt;nz-odbc-home&gt;/lib64 ln -s libnzodbc.a libnzodbc.so</td>
</tr>
</tbody>
</table>
Table 12. Connection errors (continued)

<table>
<thead>
<tr>
<th>Error type</th>
<th>Error details</th>
<th>Troubleshooting details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrect Data Source Name (DSN)</td>
<td>When running a Netezza connector DataStage job, or running a Metadata Import Wizard, or performing View data, you might encounter the following error: <code>Connection to database testdb with user testuser using DSN NZSQL failed. Reason: [SQLCODE=1M002] [Native=0] [DataDirect][ODBC lib] Data source name not found and no default driver specified</code></td>
<td>For the error in a Windows system, start Data source (ODBC) and check the DSN name. For MDI and View data check the account under which the ASBAgent service is running. If the ASBAgent service is running under the system account, the DSN names must be defined as System DSNs. Otherwise, both the system and the user DSNs can be used. An InfoSphere DataStage job can access both system and user DSNs. On a Unix system if <code>NZ_ODBC_INI_PATH</code>, is not set correctly you might encounter the following error: <code>Connection to database testdb with user testuser using DSN NZSQL failed. Reason: [SQLCODE=HY000] [Native=33] Server and/or port attributes are empty</code> For the error in a Unix system, you need to make sure that the case of the DSN matches. Also make sure that the environment variables: <code>ODBCINI</code> and <code>NZ_ODBC_INI_PATH</code> are set correctly. <code>ODBCINI</code> need to point to the correct <code>odbc.ini</code> file, and <code>NZ_ODBC_INI_PATH</code> needs to be set to the directory where the correct <code>odbc.ini</code> file is located.</td>
</tr>
<tr>
<td>User name/Password Issues</td>
<td>When running a Netezza connector DataStage job, or running a Metadata Import Wizard, or performing View data, you might encounter the following error: <code>Connection to database testdb with user testuser using DSN NZSQL failed. Reason: [SQLCODE=28000] [Native=24] Password authentication failed for user 'testuser1'</code></td>
<td>Verify that the user name/password combination is entered correctly in the job/MDI wizard. Also make sure that the specified user has the right privileges to connect to the selected database.</td>
</tr>
<tr>
<td>Firewall Issues</td>
<td>If the Netezza server is behind a firewall and you have not authenticated prior to running a Netezza connector job, or running a Metadata Import Wizard, or performing View data, you might encounter the following error: <code>Connection to database testdb with user testuser using DSN NZSQL failed. Reason: [SQLCODE=HY000] [Native=0] [DataDirect][ODBC lib] Unicode converter truncated character</code></td>
<td>Authenticate with the firewall prior to running a Netezza connector job, or running a Metadata Import Wizard, or performing View data.</td>
</tr>
</tbody>
</table>

**Runtime errors**

After the Netezza connector successfully connects to the Netezza server you might encounter the following errors:
<table>
<thead>
<tr>
<th>Error type</th>
<th>Error details</th>
<th>Troubleshooting details</th>
</tr>
</thead>
</table>
| Schema Reconciliation Issues                    | 1. Extra Link Columns  
When running a Netezza connector job, you might encounter the following error: The following input link column(s): `col2` were not found in the table `test_table`  
2. Extra Table Columns  
When running a Netezza connector job, you might encounter the following error: The following column(s): `COL2` from the table `test_table` were not found in the input link schema | The Extra Link Columns error occurs when the job is referring to a column that does not exist in the target table or the temporary work table. You need to either make sure that your job and your tables have the same columns, or configure the connector to drop the extra columns by setting the connector property: **Schema reconciliation → Unmatched link column action** to **Drop**.  
The Extra Table Columns error occurs when the target table or the temporary work table has one or more columns not found in the link schema. You need to either make sure that your job and your tables have the same columns, or instruct the connector to ignore the extra columns by setting the connector property: **Schema reconciliation → Unmatched table column action** to **Ignore nullable** or **Ignore all**. The job might still fail if the extra columns are not nullable or if they do not have a default value. |
| Temporary Work Table (TWT) is in a Separate Database | When using an existing temporary work table in a separate database, the Netezza connector job might fail with the following error:  
**Table twttable not found** | You need to make sure that the correct TWT name is specified, if a fully qualified name is given, and if all the name parts are correct. Also, make sure that the connector property **Use separate connection for TWT** is set to **Yes** and that the child properties are set to the correct values. |
Chapter 4. Environment variables: Netezza connector

The Netezza Connector stage uses these environment variables.

**CC_GUARDIUM_EVENTS**

Set this environment variable to specify whether connectors report the InfoSphere DataStage context information to the InfoSphere Guardium Database Activity monitor.

When the value of this environment variable is set, the connectors report the InfoSphere DataStage context information such as host, project, job names, stage name and node ID that the stage is running on to the InfoSphere Guardium Database Activity monitor. When this environment variable is defined and set to any value, the connectors report context information to the Guardium server after the initial connection is established.

When this environment variable is undefined, the connectors do not attempt to report context information to Guardium servers. The setting of this environment variable applies to all database connectors in the job.

**CC_IGNORE_TIME_LENGTH_AND_SCALE**

Set this environment variable to change the behavior of the connector on the parallel canvas.

When this environment variable is set to 1, the connector running with the parallel engine ignores the specified length and scale for the timestamp column. For example, when the value of this environment variable is not set and if the length of the timestamp column is 26 and the scale is 6, the connector on the parallel canvas considers that the timestamp has a microsecond resolution. When the value of this environment variable is set to 1, the connector on the parallel canvas does not consider that the timestamp has a microsecond resolution unless the microseconds extended property is set even if the length of the timestamp column is 26 and the scale is 6.

**CC_MSG_LEVEL**

Set this environment variable to specify the minimum severity of the messages that the connector reports in the log file.

At the default value of 3, informational messages and messages of a higher severity are reported to the log file.

The following list contains the valid values:

- 1 - Trace
- 2 - Debug
- 3 - Informational
- 4 - Warning
- 5 - Error
- 6 - Fatal
**CC_NZ_LOG_LEVEL**

Set this environment variable to specify the minimum severity of the messages that the connector reports in the log file.

At the default value of 3, informational messages and messages of a higher severity are reported to the log file. The following list contains the valid values:

- 1 - Trace
- 2 - Debug
- 3 - Informational
- 4 - Warning
- 5 - Error
- 6 - Fatal

This variable is identical to the **CC_MSG_LEVEL** environment variable, but except that it affects only the Netezza connector messages.

**CC_TRUNCATE_NSTRING_WITH_NULL**

Set this environment variable to truncate string data that includes the string 0x00.

When the value of this environment variable is set and when the input data contains a null character, the input data is truncated with 0x00 and the rest of the string is dropped.

**CC_USE_EXTERNAL_SCHEMA_ON_MISMATCH**

Set this environment variable to use an external schema rather than a design schema when the schemas do not match.

This schema is used for schema reconciliation. When the value of this environment variable is set, the behavior remains the same and is not changed from the old version.

**TMPDIR**

Set this environment variable to specify the directory where the Netezza connector creates named pipes to communicate with Netezza on Linux or UNIX operating systems.

This variable also specifies the directory for Netezza log files on all operating systems.
Appendix A. Product accessibility

You can get information about the accessibility status of IBM products.

The IBM InfoSphere Information Server product modules and user interfaces are not fully accessible. The installation program installs the following product modules and components:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary Anywhere
- IBM InfoSphere DataStage
- IBM InfoSphere FastTrack
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Information Services Director
- IBM InfoSphere Metadata Workbench
- IBM InfoSphere QualityStage

For information about the accessibility status of IBM products, see the IBM product accessibility information at http://www.ibm.com/able/product_accessibility/index.html.

Accessible documentation

Accessible documentation for InfoSphere Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to set display preferences in your browser. It also allows you to use screen readers and other assistive technologies to access the documentation.

The documentation that is in the information center is also provided in PDF files, which are not fully accessible.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.
Appendix B. Reading command-line syntax

This documentation uses special characters to define the command-line syntax.

The following special characters define the command-line syntax:

- [] Identifies an optional argument. Arguments that are not enclosed in brackets are required.
- ... Indicates that you can specify multiple values for the previous argument.
- | Indicates mutually exclusive information. You can use the argument to the left of the separator or the argument to the right of the separator. You cannot use both arguments in a single use of the command.
- {} Delimits a set of mutually exclusive arguments when one of the arguments is required. If the arguments are optional, they are enclosed in brackets ([ ]).

Note:
- The maximum number of characters in an argument is 256.
- Enclose argument values that have embedded spaces with either single or double quotation marks.

For example:

```
wsesrcl-S server] [-l label] [-n name] source
```

The `source` argument is the only required argument for the `wsesrc` command. The brackets around the other arguments indicate that these arguments are optional.

```
wlsac [-l | -f format] [key... ] profile
```

In this example, the `-l` and `-f format` arguments are mutually exclusive and optional. The `profile` argument is required. The `key` argument is optional. The ellipsis (...) that follows the `key` argument indicates that you can specify multiple key names.

```
wrbl-import {rule_pack | rule_set}...
```

In this example, the `rule_pack` and `rule_set` arguments are mutually exclusive, but one of the arguments must be specified. Also, the ellipsis marks (...) indicate that you can specify multiple rule packs or rule sets.
Appendix C. How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>> symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol indicates that a syntax diagram is continued from the previous line.
  - The -->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

```
>>>required_item
```

- Optional items appear below the main path.

```
>>>required_item--
optional_item
```

If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

```
>>>required_item--
optional_item
```

- If you can choose from two or more items, they appear vertically, in a stack.
  If you must choose one of the items, one item of the stack appears on the main path.

```
>>>required_item--
required_choice1
required_choice2
```

If choosing one of the items is optional, the entire stack appears below the main path.

```
>>>required_item--
optional_choice1
optional_choice2
```

If one of the items is the default, it appears above the main path, and the remaining choices are shown below.

```
>>>required_item--
default_choice
optional_choice1
optional_choice2
```

- An arrow returning to the left, above the main line, indicates an item that can be repeated.
If the repeat arrow contains a comma, you must separate repeated items with a comma.

A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.

**Fragment-name:**

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown.
- Variables appear in all lowercase italic letters (for example, `column-name`). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses, for example (1).
Appendix D. Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You also can provide feedback to IBM about products and documentation.

The following table lists resources for customer support, software services, training, and product and solutions information.

*Table 14. IBM resources*

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description and location</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Support Portal</td>
<td>You can customize support information by choosing the products and the topics that interest you at <a href="http://www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server">www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server</a></td>
</tr>
<tr>
<td>Software services</td>
<td>You can find information about software, IT, and business consulting services, on the solutions site at <a href="http://www.ibm.com/businesssolutions/">www.ibm.com/businesssolutions/</a></td>
</tr>
<tr>
<td>My IBM</td>
<td>You can manage links to IBM Web sites and information that meet your specific technical support needs by creating an account on the My IBM site at <a href="http://www.ibm.com/account/">www.ibm.com/account/</a></td>
</tr>
<tr>
<td>Training and certification</td>
<td>You can learn about technical training and education services designed for individuals, companies, and public organizations to acquire, maintain, and optimize their IT skills at <a href="http://www.ibm.com/software/sw-training/">http://www.ibm.com/software/sw-training/</a></td>
</tr>
</tbody>
</table>
Appendix E. Accessing and providing feedback on the product documentation

Documentation is provided in a variety of locations and formats, including in help that is opened directly from the product client interfaces, in a suite-wide information center, and in PDF file books.

The information center is installed as a common service with IBM InfoSphere Information Server. The information center contains help for most of the product interfaces, as well as complete documentation for all the product modules in the suite. You can open the information center from the installed product or from a Web browser.

Accessing the information center

You can use the following methods to open the installed information center.

• Click the Help link in the upper right of the client interface.

  Note: From IBM InfoSphere FastTrack and IBM InfoSphere Information Server Manager, the main Help item opens a local help system. Choose Help > Open Info Center to open the full suite information center.

• Press the F1 key. The F1 key typically opens the topic that describes the current context of the client interface.

  Note: The F1 key does not work in Web clients.

• Use a Web browser to access the installed information center even when you are not logged in to the product. Enter the following address in a Web browser: http://host_name:port_number/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.home.doc/ic-homepage.html. The host_name is the name of the services tier computer where the information center is installed, and port_number is the port number for InfoSphere Information Server. The default port number is 9080. For example, on a Microsoft® Windows® Server computer named iisdocs2, the Web address is in the following format: http://iisdocs2:9080/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.nav.doc/dochome/iisinfsrv_home.html.

A subset of the information center is also available on the IBM Web site and periodically refreshed at http://pic.dhe.ibm.com/infocenter/iisinfsv/v9r1/index.jsp.

Obtaining PDF and hardcopy documentation

• A subset of the PDF file books are available through the InfoSphere Information Server software installer and the distribution media. The other PDF file books are available online and can be accessed from this support document: https://www.ibm.com/support/docview.wss?uid=swg27008803&wv=1

• You can also order IBM publications in hardcopy format online or through your local IBM representative. To order publications online, go to the IBM Publications Center at http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss
Providing comments on the documentation

Your feedback helps IBM to provide quality information. You can use any of the following methods to provide comments:

- To comment on the information center, click the Feedback link on the top right side of any topic in the information center.
- **Send your comments by using the online readers’ comment form at** [www.ibm.com/software/awdtools/rcf/](http://www.ibm.com/software/awdtools/rcf/)
- **Send your comments by e-mail to comments@us.ibm.com. Include the name of the product, the version number of the product, and the name and part number of the information (if applicable). If you are commenting on specific text, include the location of the text (for example, a title, a table number, or a page number).**
- **You can provide general product feedback through the Consumability Survey at** [http://www.ibm.com/software/ucd/consumability/](http://www.ibm.com/software/ucd/consumability/)
Notices and trademarks

This information was developed for products and services offered in the U.S.A.

Notices

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web
sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to
IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

The following terms are trademarks or registered trademarks of other companies:

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The United States Postal Service owns the following trademarks: CASS, CASS Certified, DPV, LACS\textsuperscript{link}, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACS\textsuperscript{link} licensee of the United States Postal Service.

Other company, product or service names may be trademarks or service marks of others.
Index

C
CC_NZ_LOG_LEVEL environment variable 53
character encoding 31
client interfaces 8
Close Command 17, 24
column definitions
setting up 33, 39
command-line syntax
conventions 57
commands
syntax 57
configure data sources 12
configuring 36
Connection 16, 23
customer support
contacting 61

D
data
looking up 44
reading 32, 33
writing or loading 38
data source
creating 1, 2
Data source 16, 23
data sources on Windows
creating 2
data types
DataStage 46, 47
importing data 46
loading data 47
Netezza 46, 47
reading data 46
writing data 47
Database 16, 23
Delimiter 18, 24
design time services
generating SQL statements at design time 45
validating SQL statements at design time 46
Drop unmatched Column 18
dserv script 4

E
environment variables
Netezza connector 53
external table 7
external table workflow 9

I
installation and configuration requirements 10

J
jobs
compiling and running 48

L
legal notices 65
Linux 1
load method 7, 9, 10
Load Method 14
load methods 7
Log files 14, 22
lookup operations 44

M
mapping
data types 46, 47
metadata
importing 30

N
National Language Support (NLS) 31
Netezza connector
accessing 32
compiling and running jobs 48
configuring as a source 33
configuring as a target 39
lookup 37
normal lookup 36
overview 29
using in jobs 29
Netezza enterprise stage 12
Netezza Enterprise stage 7
Netezza identifiers
Delimited identifiers 30
Regular identifiers 30
Netezza ODBC driver
configuring 1
Netezza Performance Server 8000 7
nzload 7
nzload workflow 10

O
ODBC driver
configuring 1, 2
ODBC driver on Windows
configuring 2
Open Command 17, 23
operating systems/platforms 9
operations 37
Options 17, 23
overview 7

P
partitions 35
specifying for a write operation 43
Password 16, 23
prerequisites 10, 12
product accessibility
accessibility 55
product documentation
accessing 63
properties
defining 42
defining for a read operation 34

R
Read 10
Read Method 23
read operations
defining properties 34
setting up column definitions 33, 39
specifying data partitioning 35
reading data 32
record ordering
Configuring 41
reference links 44

S
Server 16, 23
set environment variables 11
setting environment variables for databases
setting 3, 4
software services
contacting 61
Source 22
sparse lookup 37
special characters
in command-line syntax 57
Specifying a Character Set Map for the Netezza Enterprise Stage 21, 25
Specifying a Description of the Netezza Enterprise Stage 20, 25
Specifying How the Netezza Enterprise Stage Executes 21, 25
stage operations
setting up column definitions 33, 39
support
customer 61
syntax
command-line 57

T
Table 15, 23
Target 14
Temporary work table 43
Temporary Work table (TWT) in Different Database 16

© Copyright IBM Corp. 2005, 2013