IBM InfoSphere DataStage and QualityStage
Version 9 Release 1

Connectivity Guide for Sybase Databases
Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 73.
## Contents

**Chapter 1. Configuring access to Sybase databases**  
1

- Configuring access to Sybase databases  
1
- Permissions required to access Sybase databases  
2
- Setting the library path environment variable  
2
  - Setting the library path environment variable in the denv file  
3
  - Setting the library path environment variable in Windows  
4
- Testing database connections by using the ISA Lite tool  
4

**Chapter 2. Introduction**  
5

- Sybase OC stages  
5
- Sybase BCP Stages  
6
- Sybase enterprise stages  
6
- Sybase IQ12 load stages  
6
- Configuring the environment for Sybase OC and Sybase IQ12 load stages  
6
- Configuring the environment for BCPLoad stages  
7
- Configuring the environment for Sybase enterprise stages  
8
- Sybase stages and the parallel canvas  
9

**Chapter 3. Sybase OC stages**  
11

- Defining the Sybase OC stage  
11
- Configuring the Sybase OC stage  
11
- Connecting to a Sybase database  
12
- Defining character set mapping  
13
- Defining Sybase input data  
13
  - About the Input page  
13
  - Writing data to Sybase  
15
- Defining Sybase output data  
17
  - About the Output page  
17
  - Reading data from Sybase  
19
- Sybase SQL server data type support  
20
  - Character data types  
20
  - Numeric (integer) data types  
21
  - Numeric (decimal) data types  
21
  - Numeric (money) data type  
22
  - Numeric (approximate) data types  
22
  - Date data types  
23
  - Binary data types  
23
- Stored procedure support  
24

**Chapter 4. BCPLoad stages**  
25

- BCPLoad stage overview  
25
- Table definitions  
26
- SQL data types  
26
- Using BCPLoad stages  
26
  - Must do’s for Sybase BCPLoad stages  
27
  - Editing stage properties  
27
  - Using stored procedures  
28
- Defining character set maps  
30
- Defining BCPLoad input data  
31

**Chapter 5. Sybase enterprise stage**  
33

- Prerequisites  
33
- Sybase enterprise stages  
33
- Environment variables  
34
- Loading data to remote Sybase IQ server  
34

**Chapter 6. Sybase enterprise stage editor**  
35

- Stage page  
35
- Input page  
36
  - General tab on the input page  
36
  - Properties tab on the input page  
36
- Setting the Sybase enterprise stage properties for the input link  
38
- Columns tab on the input page  
42
- Advanced tab on the input page  
42
- Output page  
42
  - General tab on the output page  
42
  - Properties tab on the output page  
42
- Setting the Sybase enterprise stage properties for the output link  
43
- Columns tab on the output page  
44
- Advanced tab on the output page  
44

**Chapter 7. Working with Sybase databases**  
45

- Using Sybase open client  
45
- Updating a Sybase database  
45
- Loading a Sybase database  
46
- Reading a Sybase database  
47
  - Reading a remote Sybase IQ server  
48
- Performing a lookup on a Sybase database  
49
  - Performing a direct lookup on a Sybase database table  
49
  - Performing an in-memory lookup on a Sybase database table  
49

**Chapter 8. Sybase IQ12 load stages**  
51

- Functionality of the Sybase IQ12 load stage  
51
- Terminology  
52
  - Sybase IQ12 terminology  
52
  - Sybase IQ12 load terminology  
52
- Index sets  
53
  - Loading index sets  
53
  - Loading joined index sets  
53
- Loading and synchronizing joined index sets  
54
- Disk overflow handling  
54
- Stage and link properties  
55
  - Stage properties  
55
  - Link properties  
58
Chapter 1. Configuring access to Sybase databases

To configure access to Sybase 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 Sybase databases.

Configuring access to Sybase databases

The Sybase Open Client environment is installed and configured on the InfoSphere® Information Server engine tier.

Before you begin
- Install and configure the SQL Server or Sybase client software. The BCPLoad stage uses the BCP API in the DBLIB/CTLIB and NetLIB client libraries. You must ensure that these components are installed on the InfoSphere DataStage® server that set up as a client to the SQL Server DBMS. For more information, see DBMS documentation.
- Make sure that the Sybase database server is accessible from the Sybase client.
- Create table in the database on the SQL Server.
- Make sure that the database is registered on the Sybase client.

Procedure
1. Set the SYBASE environment variable to point to the Sybase installation directory. For example, export SYBASE=/disk3/Sybase.
2. Set the SYBASE_OCS environment variable to point to the Sybase Open Client directory. For example, export SYBASE_OCS=OCS-12.5. This value indicates the version and release of the Open Client product.
3. Specify the database name, host system name or IP address, and port number in the interfaces file (for example, sql.ini) in the $SYBASE directory.
4. Set the DSQUERY environment variable to the name of the Sybase database server to connect to by default when the server name is not specified in the connection request. If the environment variable is not set, the default value SYBASE is used.
5. Set the PATH and library path environment variable to point to the directory that contains the client libraries.
   - On Windows, %SYBASE%\%SYBASE_OCS%\bin and %SYBASE%\%SYBASE_OCS%\dll directories, where SYBASE and SYBASE_OCS represent the Sybase product installation home directory and Sybase Open Client directory.
   - On Linux and UNIX, $SYBASE/$SYBASE_OCS/bin.

   Note: Make sure that $SYBASE/$SYBASE_OCS/bin is displayed first in the PATH environment variable.
6. For the BCPLoad stages, configure the database for the fast copy (bulk load) option, by using a stored procedure. When this option is used, data is loaded
without recording every insert in a log file. For more information about using stored procedures, see Using Stored Procedures.

7. Get login privileges to Sybase by using a valid Sybase user name and corresponding password, server name, and database. These must be recognized by Sybase before you attempt to access it.

8. Use the dsedit utility that is provided with the Sybase Open Client to configure connection to the Sybase database.

9. Test the connectivity to Sybase database outside of InfoSphere DataStage by using tools such as $SYBASE/$SYBASE_OCS/bin/isql tool in Sybase Open Client.

10. Complete the following steps to access Sybase databases with NLS in Sybase enterprise stages.
   a. Create a database and configure the language of your choice. For example, create database <<database path>> COLLATION 932JPN for a Japanese (shift_jis) database.
   b. Install the IBM® InfoSphere DataStage server in that particular language, for example, Japanese (shift_jis).
   c. To set the language for InfoSphere DataStage client, use the NLS tab in job properties to select the language.
   d. Make sure that the selected language is set as default in the operating system of the system on which the InfoSphere DataStage client is installed.

**Permissions required to access Sybase databases**

To complete operations on tables that are hosted by Sybase ASE and Sybase IQ databases, you require specific privileges on the table.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Options</th>
<th>Syspartition (only for Sybase ASE)</th>
<th>sysobjects</th>
<th>SELECT privilege on table</th>
<th>INSERT privilege on table</th>
<th>Delete table</th>
<th>Create table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write</td>
<td>Create/Replace</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Write</td>
<td>Append</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Write</td>
<td>Truncate</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Read</td>
<td>All</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Upset</td>
<td>Update/Insert</td>
<td>To complete an update operation, you require the UPDATE privilege on the table that you want to update. To complete an insert operation, you require the INSERT privilege on the table into which you want to insert records.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Lookup</td>
<td>All</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

**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 operating systems, and HP-IA</td>
<td>LD_LIBRARY_PATH</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. Introduction

IBM InfoSphere DataStage provides several ways to access Sybase databases. The following topics provide an introduction to the stages that access Sybase databases. Later topics cover installation instructions and configuration information.

With the InfoSphere DataStage, you can load tables in Sybase databases and read data from tables in Sybase databases. This guide describes how to use

- Sybase OC stages
- Sybase BCP stages
- Sybase Enterprise stages
- Sybase IQ12 Load stages

in the InfoSphere DataStage server jobs.

These connectivity stages are installed automatically when you install the InfoSphere DataStage. They appear in the Database category on the server job palette.

Although these topics contains several stages specifically designed to access Sybase databases, you can also access them by using the following stages:

- Dynamic Relational Stage (DRS)
  Use this stage to create a job that requires one relational database at design time and another at runtime. For more information about this stage, see the Connectivity Guide for the Dynamic Relational Stage.

- Stored Procedure (STP)
  Use this stage to include Sybase stored procedures as part of your InfoSphere DataStage job. For more information about this stage, see the Connectivity Guide for the Dynamic Relational Stage.

Sybase OC stages

Sybase Open Client allows C and C++ applications to connect to and process SQL statements in the Sybase SQL Server environment. Sybase OC enables InfoSphere DataStage to read and write data to and from a Sybase database by using the Sybase Open Client Client-Library interface.

Each Sybase OC stage is a passive stage that can have any number of the following links:

- **Input links.** Specify the data you are writing, which is a stream of rows to be loaded into a Sybase database.

- **Output links.** Specify the data you are extracting, which is a stream of rows to be read from a Sybase database. You can specify the data on an input or output link by using an SQL statement constructed by InfoSphere DataStage or a user-defined query.

- **Reference output link.** Each represents rows that are key read from a Sybase database (by using the key columns in a WHERE clause of the SELECT statement that is constructed by InfoSphere DataStage or specified by the user).

For SQL syntax information, see your database documentation.
**Sybase BCP Stages**

Microsoft SQL Server and Sybase have a utility called BCP (Bulk Copy Program). This command line utility copies SQL Server data to or from an operating system file in a user-specified format. BCP uses the bulk copy API in the SQL Server client libraries.

By using BCP, you can load large volumes of data into a table without recording each insert in a log file. You can run BCP manually from a command line by using the command line options (switches). A format (.fmt) file is created which is used to load the data into the database.

The BCPLoad stage uses the BCP (Bulk Copy Program) utility to bulk load data into a single table in a Microsoft SQL Server 2000 or Sybase (System 11.5 or 12.5) database.

**Sybase enterprise stages**

The Sybase enterprise Stage allows you to read data from and write data to a Sybase database. This stage is not available as a server stage.

**Sybase IQ12 load stages**

The Sybase Adaptive Server IQ for Version 12 (Sybase IQ12) is an advanced indexing engine, not a database management system. The goal of the Sybase IQ12 Load stage is to enable InfoSphere DataStage Release 7.0 to rapidly and efficiently load data into existing Sybase IQ index sets.

Each input link to the stage represents a stream of rows to load into a Version 12 Sybase Adaptive Server IQ index set or joined index set. Reference links and output links have no meaning in the context of the Sybase IQ12 stage and are not allowed.

The Sybase IQ12 Load is not a migration tool. Use Sybase tools to generate scripts if you need to migrate data from earlier versions of the Sybase Adaptive Server IQ. Consult the Sybase Adaptive Server IQ Version 12 documentation for help with migration issues.

**Configuring the environment for Sybase OC and Sybase IQ12 load stages**

**About this task**

Some of the Sybase stages require the setting of environment variables in order to work correctly on a UNIX platform. To add or change an environment variable, include any environment variables in the dsenv file. The table below identifies the specific environment variables that are required.

<table>
<thead>
<tr>
<th>For...</th>
<th>Set the following environment variable on the IBM InfoSphere DataStage server machine...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybase OC</td>
<td>SYBASE SYBASE_OCS</td>
</tr>
<tr>
<td></td>
<td>LD_LIBRARY_PATH</td>
</tr>
</tbody>
</table>

6 Connectivity Guide for Sybase Databases
Table 2. Required environment variables for UNIX (continued)

<table>
<thead>
<tr>
<th>For...</th>
<th>Set the following environment variable on the IBM InfoSphere DataStage server machine...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybase IQ12 Load</td>
<td>SYBASE ASDIR SYBASE_OCS LD_LIBRARY_PATH</td>
</tr>
</tbody>
</table>

1To run successfully on Linux, Sybase OC requires the language environment variable LANG = en.

2In Sybase OC applications, for the LD_LIBRARY_PATH environment variable, the InfoSphere DataStage library entries must be referenced before the Sybase Open Client library entries at run time.

3The name of one particular environment variable, referred to as LD_LIBRARY_PATH above, differs depending on the platform. To determine the correct name to use for your environment.
   - If the platform is AIX, use LIBPATH.
   - If the platform is HP_UX, use SHLIB_PATH.
   - If the platform is LINUX, Solaris, or Tru64, use LD_LIBRARY_PATH.

Note: If you experience any timeout issues, increase the default values for the Sybase configuration parameters CS_RETRY_COUNT and CS_TIMEOUT_VALUE to 10 or higher.

Configuring the environment for BCPLoad stages

About this task

Before you can use the BCPLoad stage you must:
   - Install and configure the SQL Server or Sybase client software. The BCPLoad stage uses the BCP API in the DBLIB/CTLIB and NetLIB client libraries. You must ensure that these components are installed on the IBM InfoSphere DataStage server that acts as a client to the SQL Server DBMS. See the documentation supplied with your DBMS for more details.
   - Use one of the client tools (for example, ISQLW in the case of Microsoft SQL Server or WISQL32 for Sybase) to ensure that the connectivity between the InfoSphere DataStage server and the SQL Server host is operational.
   - Create the table in the database on the SQL Server.
   - Configure your database to use the fast copy (bulk load) option. By using this option, the data is loaded without each insert being recorded in a log file. If you do not specify this setting, all transactions are logged, slowing down the rate at which data is loaded. The fast copy option can be switched on by a stored procedure. For more information about using stored procedures, see Using Stored Procedures.

There are some special points to note about SQL Server. If the following error is returned when you are using the BCPLoad stage with data in YMD format, and the Date Format has been set:

Attempt to convert data stopped by syntax error in source field. If your table contains date fields in ymd format make sure the Date Format property is set...
then clear the **Use International Settings** check box in the DB-Library option page of the SQL Server Client Network Utility.

If your job uses data in the upper 128 characters of the character set and the data is not appearing correctly on the database then clear the **Automatic ANSI to OEM conversion** check box in the DB-Library option page of the SQL Server Client Network Utility.

---

### Configuring the environment for Sybase enterprise stages

#### About this task

You must install the Sybase open client software on the server side for the Sybase Enterprise Stage to function. The configuration details are below.

- Create the user-defined environment variable SYBASE and set this to the $SYBASE path that specifies the Sybase home directory (for example, export SYBASE=/disk3/Sybase).
- Create the user-defined environment variable SYBASE_OCS and set this to the Sybase open client software installation directory (for example, export SYBASE_OCS=OCS-12_5).
- Interfaces file: Add the details about the database server (database name, host machine name or IP address, and port number) to the interfaces file located in $SYBASE directory.
- Add $SYBASE/$SYBASE_OCS/bin to your PATH and $SYBASE/$SYBASE_OCS/lib to your LIBPATH, LD_LIBRARY_PATH, or SHLIB_PATH.
- Get login privileges to Sybase by using a valid Sybase user name and corresponding password, server name, and database. These must be recognized by Sybase before you attempt to access it.

**Note:** $SYBASE/$SYBASE_OCS/bin must appear first in your PATH. This is to ensure that $SYBASE/$SYBASE_OCS/bin/isql is executed whenever the user executes the “isql” command.

The steps for accessing Sybase databases with NLS are as follows:

#### Procedure

1. Create a database and configure the language that you wish to test for this database. (For example, create database <<database path>> COLLATION 932JPN for a Japanese (shift_jis) database.
2. Install the IBM InfoSphere DataStage server in that particular language, for example, Japanese (shift_jis). Upgrading the existing InfoSphere DataStage server will not work as you will not get any option for selecting another language. You must uninstall and reinstall the existing server in the desired language.
3. Make sure that the language that you want to test is a default setting in the operating system of the machine which you will use to test through the InfoSphere DataStage client. Select the appropriate language by using the **Control Panel > Regional Settings** option in the **Start** menu on your desktop. Additionally, you must set the keyboard input to the language that you want to test.
Results

For the InfoSphere DataStage client setting, use the NLS tab in the Sybase Enterprise stage to select the language that you want to test. For example, if Japanese is the default language for your operating system, then the project default in the InfoSphere DataStage client will be Shift_JIS. You do not need to select the language for every job that you run.

Sybase stages and the parallel canvas

Some connectivity stages can run on the parallel canvas. The default for all stages is Sequential. "In Parallel" means you can set it to run in parallel, but this is NOT the default. Use the table below to determine which connectivity stages are available on the parallel canvas.

Table 3. Availability on the Parallel Canvas

<table>
<thead>
<tr>
<th>Connectivity Stage</th>
<th>Available on the Parallel Canvas in Windows</th>
<th>Available on the Parallel Canvas in UNIX</th>
<th>Used as a Source or a Target or for Processing</th>
<th>Runs Sequentially or In Parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybase OC</td>
<td>Yes</td>
<td>Yes</td>
<td>Source or Target</td>
<td>Source: Sequential; Target: In Parallel</td>
</tr>
<tr>
<td>Sybase IQ12 Load</td>
<td>Yes</td>
<td>Yes</td>
<td>Target</td>
<td>Sequential</td>
</tr>
</tbody>
</table>

BCPLoad cannot run on the parallel canvas.

Sybase enterprise stage is not a server stage and runs exclusively on the parallel canvas.
Chapter 3. Sybase OC stages

This topic describes the following subjects for the Sybase OC stage:

- “Defining the Sybase OC stage”
- “Defining character set mapping” on page 13
- “Defining Sybase input data” on page 13
- “Defining Sybase output data” on page 17
- “Sybase SQL server data type support” on page 20
- “Stored procedure support” on page 24

Defining the Sybase OC stage

To edit a Sybase OC stage, open the SYBASEOC stage dialog box. This dialog box has the following pages (depending on whether there are inputs to and outputs from the stage):

- **Stage.** This page displays the name of the stage you are editing. The **General** tab defines the Sybase server, database, login information, transaction isolation level, and packet size information for concurrency control and performance tuning in jobs. You can connect to a Sybase database. You can describe the purpose of the stage in the **Description** field on the **General** tab. For details, see “Connecting to a Sybase database” on page 12.

  The **NLS** tab defines a character set map to be used with the stage. (This tab appears only if you have installed NLS.) For details, see “Defining character set maps” on page 30.

- **Input.** This page is displayed only if you have an input link to this stage. It specifies the SQL table to use and the associated column definitions for each data input link. This page also specifies how data is written and contains the SQL statement or call syntax used to write data to a Sybase table.

- **Output.** This page is displayed only if you have an output or reference link to this stage. It specifies the SQL tables to use and the associated column definitions for each data output link. It also contains the SQL SELECT statement or call syntax used to read data from one or more Sybase tables or views.

Defining the Sybase OC stage

About this task

To define the Sybase OC stage by using the SYBASEOC Stage dialog box:

**Procedure**

1. Connect to an Sybase database (see the next topic).
2. Optionally define a character set map (see “Defining character set mapping” on page 13).
3. Define the data on the input links (see “Defining Sybase input data” on page 13).
4. Define the data on the output links (see “Defining Sybase output data” on page 17).
Connecting to a Sybase database

About this task

The Sybase OC connection parameters are set on the General tab of the Stage page. To connect to a Sybase database:

Procedure

1. Enter the name of the system where the Sybase server is installed in the Server field. This name should correspond to an entry in the Sybase sql.ini (Windows clients) or interfaces (UNIX clients) file that contains the server information. This field is required. There is no default.

2. Enter the name of the Sybase database name to access in the Database field. Unless the database has a guest account, User must be a valid user in the database, have an alias in the database, or be a system administrator or system security officer. If you do not specify Database, the default database for the user (as configured in Sybase) is accessed.

3. Enter the name to use to connect to the Sybase server in the User field. This user must have sufficient privileges to access the specified database and source and target tables. This field is required. There is no default.

4. Enter the password that is associated with the specified user name to use in the Password field. There is no default.

5. Choose an appropriate transaction isolation level to use from the Transaction Isolation list. This level provides the necessary concurrency control between transactions in the job and other transactions.

6. Use one of the following transaction isolation levels:

   **Level 0 (read uncommitted).** Takes exclusive locks on modified data. These locks are held until a commit or rollback is executed. However, other transactions can still read (but not modify) the uncommitted changes. No other locks are taken.

   **Level 1 (read committed).** Takes exclusive locks on modified data and sharable locks on all other data. Exclusive locks are held until a commit or rollback is executed. Uncommitted changes are not readable by other transactions. Shared locks are released immediately after the data has been processed, allowing other transactions to modify it.

   **Level 3 (serializable).** Takes exclusive locks on modified data and sharable locks on all other data. All locks are held until a commit or rollback is executed, preventing other transactions from modifying any data that has been referenced during the transaction.

7. Enter the packet size for Sybase client/server communication in the Packet size field. The correct setting of this property can improve performance.

   The value must be a multiple of 512 and cannot exceed the "max network packet size" Sybase server parameter. If the specified value is not a multiple of 512, it is rounded down to the nearest multiple. If no value is specified, the "default network packet size" server parameter value is used.

   To increase the "max network packet size" parameter or check its current value, use the Sybase sp_configure system procedure. You must also increase the "additional network memory" server parameter to accommodate any increase in "max network packet size." After changing these parameters, you must restart the Sybase server so the new values take effect.
For more information on setting server parameters, see your Sybase documentation. For information on the performance advantages of increasing the "max network packet size" server parameter, see your Sybase documentation.

8. Optionally describe the Sybase OC stage in the Description field.

**Defining character set mapping**

**About this task**

You can define a character set map for a stage from the NLS tab that appears on the Stage page. The NLS tab appears only if you have installed NLS.

The default character set map is defined for the project or the job. You can change the map by selecting a map name from the Map name to use with stage list.

Click Use Job Parameter... to specify parameter values for the job. Use the format #Param#, where Param is the name of the job parameter. The string #Param# is replaced by the job parameter when the job is run.

Select Show all maps to list all the maps that are shipped with InfoSphere DataStage.

Select Loaded maps only to list only the maps that are currently loaded.

**Defining Sybase input data**

When you write data to a table in a Sybase database, the Sybase OC stage has an input link. The properties of this link and the column definitions of the data are defined on the Input page in the SYBASEOC Stage dialog box.

**About the Input page**

The Input page has one field, three tabs, and two buttons:

- **Input name.** The name of the input link. Choose the link you want to edit from the Input name list. This list displays all the input links to the Sybase OC stage.
- **General.** This tab is displayed by default. It contains the following parameters:
  - **Table name.** This field is editable when the update action is not User defined SQL (otherwise, it is read-only). Depending on the update action selected, the user name specified on the Stage page must have insert, update, or delete privileges on the table named in Table name.
    - It is the name of the target table the data is written to. You must specify Table name if you do not specify User defined SQL. There is no default.
    - You can also click ... to browse the Repository to select the table.
  - **Update action.** Specifies which SQL statements are used to update the target table. Some update actions require key columns to update or delete rows. There is no default. Choose the option you want from the list:
    - **Clear table then insert rows.** Clears the table by using the delete command and inserts new rows. Delete is equivalent to, but slower than, truncate table. (Delete removes rows one at a time and logs each deleted row as a transaction. Truncate table deallocates whole data pages and makes fewer log entries.)
    - **Truncate table then insert rows.** Clears the table by using the truncate table command and inserts new rows. Truncate table is equivalent to, but faster
than, delete. (Delete removes rows one at a time and logs each deleted row as a transaction. Truncate table deallocates whole data pages and makes fewer log entries.)

**Insert rows without clearing.** Inserts new rows in the table.

**Delete existing rows only.** Deletes the existing rows in the target file that have identical keys in the source files.

**Replace existing rows completely.** Deletes the existing rows, then adds the new rows to the table.

**Update existing rows only.** Updates the existing data rows. Any rows in the data that do not exist in the table are ignored.

**Update existing rows or insert new rows.** Updates the existing data rows before adding new rows. Performance depends on the contents of the target table and the rows being processed in the job. If most rows exist in the target table, it is faster to update first.

**Insert new rows or update existing rows.** Inserts the new rows before updating existing rows. Performance depends on the contents of the target table and the rows being processed in the job. If most rows do not exist in the target table, it is faster to insert first.

**User defined SQL.** Writes the data by using a user-defined SQL statement. When you select this option, it overrides the default SQL statement generated by the stage.

- **Transaction size.** The number of rows that the stage processes before committing a transaction to the database. The default value of 100 is recommended for optimal performance. If set to 0, the Sybase OC stage commits one transaction after all rows are processed. Setting this property to 0 or a large nonzero value requires Sybase to maintain very large open transactions, which adds overhead and can decrease performance. Setting this property to a small nonzero value results in frequent transaction commits, which also adds overhead and can decrease performance.

- **Description.** Contains an optional description of the input link.

- **Columns.** This tab contains the column definitions for the data written to the table or file. The column definitions are used in the order they appear in the Columns grid. The **Columns** tab behaves the same way as the **Columns** tab in the ODBC stage.

- **SQL.** This tab contains one field and four tabs:
  - **Input name.** The name of the input link. Choose the link you want to edit from the **Input name** list. This list displays all the input links to the Sybase OC stage.
  - **Generated.** This tab is displayed by default. It contains the SQL statements constructed by the Sybase OC stage that are used to write data to Sybase. You cannot edit these statements, but you can use **Copy** to copy them to the clipboard for use elsewhere. See “Using generated SQL statements” on page 15.
  - **User-defined.** This tab contains the SQL statements executed to write data to Sybase. See “Using user-defined SQL statements” on page 16.
  - **Before.** This tab contains the SQL statements executed before the stage processes any job data rows. See “Using BeforeSQL statements” on page 16.
  - **After.** This tab contains the SQL statements executed after the stage processes job data rows. See “Using AfterSQL statements” on page 17.

Click **Columns...** to display a brief list of columns designated on the input link. As you enter detailed metadata in the **Columns** tab, you can leave this list displayed.
Click View Data... to start the Data Browser. This lets you look at the data associated with the input link.

**Writing data to Sybase**

The following sections describe the differences when you use generated or user-defined SQL INSERT, DELETE, or UPDATE statements to write data from the Sybase OC stage to a Sybase database.

**Using generated SQL statements**

**About this task**

By default, the Sybase OC stage writes data to a Sybase table by using an SQL INSERT, DELETE, or UPDATE statement that it constructs. The generated SQL statement is automatically constructed by using the table and column definitions that you specify in the input properties for this stage. The Generated tab on the SQL tab displays the SQL statement used to write the data.

To use a generated statement:

**Procedure**

1. Enter a table name in the Table name field on the General tab of the Input page.
2. Specify how you want the data to be written by choosing an option from the Update action list:
   - Clear table then insert rows
   - Truncate table then insert rows
   - Insert rows without clearing
   - Delete existing rows only
   - Replace existing rows completely
   - Update existing rows only
   - Update existing rows or insert new rows
   - Insert new rows or update existing rows
   - User defined SQL
     See “Defining Sybase input data” on page 13 for a description of each update action.
3. Enter an optional description of the input link in the Description field.
4. Click the Columns tab on the Input page.
5. Edit the Columns grid to specify column definitions for the columns you want to write.
   The SQL statement is automatically constructed by using your chosen update action and the columns you have specified. You can now optionally view this SQL statement.
6. Click the SQL tab on the Input page, then the Generated tab to view this SQL statement. You cannot edit the statement here, but you can access this tab at any time to select and copy parts of the generated statement to paste into the user-defined SQL statement.
7. Click OK to close the SYBASEOC Stage dialog box. Changes are saved when you save your job design.
Using user-defined SQL statements

About this task

Instead of writing data by using an SQL statement constructed by the stage, you can enter your own SQL INSERT, DELETE, or UPDATE statement for each Sybase OC input link. Ensure that the SQL statement contains the table name, the type of update action you want to perform, and the columns you want to write.

To enter an SQL statement:

Procedure

1. Choose User defined SQL from the Update action list on the General tab of the Input page.
2. Click the User-defined tab on the SQL tab. The User-defined SQL tab opens. Enter the SQL statement you want to use or edit to write data to the target Sybase tables. This statement must contain the table name, the type of update action you want to perform, and the columns you want to write.

If the property value begins with {FILE}, the remaining text is interpreted as a pathname, and the contents of the file supplies the property value.

When writing data, the INSERT statements must contain a VALUES clause with a parameter marker (?) for each stage input column. UPDATE statements must contain a SET clause with parameter markers for each stage input column. UPDATE and DELETE statements must contain a WHERE clause with parameter markers for the primary key columns. The parameter markers must be in the same order as the associated columns listed in the stage properties. For example:

```
insert emp (emp_no, emp_name) values (?, ?)
```

If you specify multiple SQL statements, they are executed as one or more Transact-SQL command batches by using a semicolon (;) as the end-of-batch signal. You cannot combine multiple INSERT, UPDATE, and DELETE statements in one batch. You must execute each in a separate command batch.

You cannot call stored procedures as there is no facility for passing the row values as parameters. (You can call stored procedures for output.)

Unless you specify a user-defined SQL statement, the stage automatically generates an SQL statement.

3. Click OK to close the SYBASEOC Stage dialog box. Changes are saved when you save your job design.

Using BeforeSQL statements

About this task

You can execute SQL statements before the stage processes any job data rows. To specify SQL statements before processing any data:

Procedure

1. Enter the SQL statements you want to be executed before data is processed in the text entry area on the Before tab of the SQL tab.

If the property value begins with {FILE}, the remaining text is interpreted as a pathname and the content of the file supplies the property value.

Execution occurs immediately after a successful connection between the Sybase client and server. If you specify multiple SQL statements, they are executed as one or more Transact-SQL command batches by using a semicolon (;) as the end-of-batch signal.
2. Select the **Treat errors as non-fatal** check box to log BeforeSQL execution errors as warnings. Processing continues with the next command batch, if any. Each successful execution is committed as a separate transaction.

   If this check box is cleared, BeforeSQL execution errors are considered fatal to the job and result in a transaction rollback. The transaction is committed only if all BeforeSQL statements successfully execute.

**Using AfterSQL statements**

**About this task**

You can execute SQL statements after the stage processes all job data rows. To specify SQL statements after processing data:

**Procedure**

1. Enter the SQL statements you want to be executed after the data is processed in the text entry area on the **After** tab.

   If the property value begins with [FILE], the remaining text is interpreted as a pathname and the content of the file supplies the property value.

   Execution occurs immediately before the connection between the Sybase client and server is terminated. If you specify multiple SQL statements, they are executed as one or more Transact-SQL command batches by using a semicolon ( ; ) as the end-of-batch signal.

2. Select the **Treat errors as non-fatal** check box to log AfterSQL execution errors as warnings. Processing continues with the next command batch, if any. Each successful execution is committed as a separate transaction.

   If this check box is cleared, AfterSQL execution errors are considered fatal to the job and result in a transaction rollback. The transaction is committed only if all AfterSQL statements successfully execute.

**Defining Sybase output data**

When you read data from a Sybase data source, the Sybase OC stage has an output link. The properties of the output link and the column definitions of the data are defined on the Output page in the SYBASEOC Stage dialog box.

**About the Output page**

The Output page has one field, up to four tabs, and two buttons. The tabs displayed depend on how you choose to specify the SQL statement to output the data.

- **Output name.** The name of the output link. Choose the link you want to edit from the **Output name** list. This list displays all the output links from the Sybase OC stage.

- **General.** This tab is displayed by default. It contains the following parameters:

  - **Table names.** This field appears only when you select **Generated SQL query.** It contains the names of the Sybase source tables or files being accessed. These tables must exist or be created and populated by the BeforeSQL statements. Separate multiple table names by a comma ( , ). You must have select privileges on each table. There is no default.

If you specify the query type as **User-defined SQL query,** **Table names** is ignored. You must specify **Table names** if you do not define the query type as **User-defined SQL query.**

You can click ... to browse the Repository to select tables.
Additionally, you can use a job parameter to specify the table name.

- **Prefetch rows.** The number of rows that Sybase returns when the stage fetches data from the source tables. Specifying a value greater than 1 improves performance (memory usage increases to accommodate buffering multiple rows). The stage uses this property to bind fetch buffer arrays. For more information, see your Sybase documentation.

- **Query type.** Displays the following options:
  - **Generated SQL query.** This is the default setting, which specifies that the data is extracted by using an SQL statement constructed by the Sybase OC stage. When this option is selected, the **Generated** tab appears on the **SQL** tab. You cannot edit this statement.
  - **User-defined SQL query.** Specifies that the data is extracted by using a user-defined SQL query. When this option is selected, the **User-defined** tab appears on the **SQL** tab allowing you to edit SQL statements.

- **Use column derivation fields.** Specifies that the column derivation field is to be used when generating the SQL SELECT statement. If **Use column derivation fields** is selected, the column derivation field is used. If **Use column derivation fields** is not selected (the default), the column derivation field is not used.

- **Description.** Contains an optional description of the output link.

- **Columns.** This tab contains the column definitions for the data being output on the chosen link. This tab also specifies which columns are aggregated.

- **Selection.** This tab is used primarily with generated SQL statements. It contains optional SQL SELECT clauses for the conditional extraction of data.

- **SQL.** This tab displays the SQL statements or stored procedure call syntax used to read data from Sybase. It contains one field and four tabs:
  - **Generated.** This tab is displayed by default. It contains the SQL statements constructed by the Sybase OC stage. You cannot edit these statements, but you can use **Copy** to copy them to the Clipboard for use elsewhere. See “Using generated queries” on page 19.
  - **User-defined.** This tab contains the SQL statements executed to write data to Sybase. This tab is enabled when you select **User-defined SQL query** from the **General** tab of the Output page. See “Using user-defined queries” on page 20.
  - **Before.** This tab contains the SQL statements executed before the stage processes any job data rows. See “Using BeforeSQL statements” on page 16.
  - **After.** This tab contains the SQL statements executed after the stage processes all job data rows. See “Using AfterSQL statements” on page 17.

Click **Columns...** to display a brief list of columns designated on the output link. As you enter detailed metadata in the **Columns** tab, you can leave this list displayed.

Click **View Data...** to start the Data Browser. This lets you look at the data associated with the output link.
**Reading data from Sybase**

The following sections describe the differences when you use generated queries or user-defined queries to read data from a Sybase database into the Sybase OC stage.

The column definitions for reference links must contain a key field. You use key fields to join primary and reference inputs to a Transformer stage. The Sybase OC plug-in key reads the data by using a WHERE clause in the SQL SELECT statement.

**Using generated queries**

By default, the stage extracts data from a Sybase OC data source by using an SQL SELECT statement that it constructs. The SQL statement is automatically constructed by using the table and column definitions that you entered on the Output page.

When you select **Generated SQL query**, data is extracted from a Sybase database by using an SQL SELECT statement constructed by the Sybase OC stage. SQL SELECT statements have the following syntax:

```
SELECT clause FROM clause
[WHERE clause]
[GROUP BY clause]
[HAVING clause]
[ORDER BY clause];
```

When you specify the tables to use and the columns to be output from the Sybase OC stage, the SQL SELECT statement is automatically constructed and can be viewed by clicking the **SQL** tab on the Output page.

For example, if you extract the columns **Name**, **Address**, and **Phone** from a table called Table1, the SQL statement displayed on the **SQL** tab is:

```
SELECT Name, Address, Phone FROM Table1;
```

The SELECT and FROM clauses are the minimum required and are automatically generated by the Sybase OC stage. However, you can use any of these SQL SELECT clauses:

- **SELECT.** Specifies the columns to select from the database.
- **FROM.** Specifies the tables containing the selected columns.
- **WHERE.** Specifies the criteria that rows must meet to be selected.
- **GROUP BY.** Groups rows to summarize results.
- **HAVING.** Specifies the criteria that grouped rows must meet to be selected.
- **ORDER BY.** Sorts selected rows.

If you want to use the additional SQL SELECT clauses, you must enter them on the **Selection** tab on the Output page.

The **Selection** tab is divided into two areas (panes). You can resize an area by dragging the split bar.

- **WHERE clause.** This text box allows you to insert an SQL WHERE clause to specify criteria that the data must meet before being selected.
- **Other clauses.** This text box allows you to insert a HAVING or an ORDER BY clause.
Using user-defined queries

About this task

Instead of using the SQL statement constructed by the Sybase OC stage, you can enter your own SQL statement for each Sybase OC output link. To enter an SQL statement:

Procedure

1. Select User-defined SQL query from the Query type list on the General tab of the Output page. The User-defined tab on the SQL tab is enabled.

You can edit the statements or drag the selected columns into your user-defined SQL statement. You must ensure that the table definitions for the output link are correct and represent the columns that are expected. The result set generated from this statement returns at least one row.

If the property value begins with {FILE}, the remaining text is interpreted as a pathname, and the content of the file supplies the property value.

The SQL statement must generate a result set that matches the stage output column definitions.

If you specify multiple SQL statements, they are executed as one or more Transact-SQL command batches by using a semicolon (;) as the end-of-batch signal.

Note: If more than one result set is produced, only the first set is used.

2. Click OK to close this dialog box. Changes are saved when you save your job design.

---

Sybase SQL server data type support

About this task

The following tables document the support for Sybase SQL server data types. When creating InfoSphere DataStage table definitions for a Sybase table, set the SQL type, length, and scale attributes as noted.

Character data types

The following table summarizes character data types for Sybase SQL server, their InfoSphere DataStage SQL type definitions, and the corresponding length attributes that you need to set.

*Table 4. Character data types*

<table>
<thead>
<tr>
<th>Sybase Data Type</th>
<th>InfoSphere DataStage SQL Type</th>
<th>Length</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>char(n)</code></td>
<td>Char</td>
<td>$n$</td>
<td>Sybase <code>char</code> values are blank padded to $n$ characters.</td>
</tr>
</tbody>
</table>
Table 4. Character data types (continued)

<table>
<thead>
<tr>
<th>Sybase Data Type</th>
<th>InfoSphere DataStage SQL Type</th>
<th>Length</th>
<th>Notes</th>
</tr>
</thead>
</table>
| nchar(n)         | Char                           | n * @ncharsize | Sybase nchar values are blank padded to n characters.  
|                  |                                |        | @ncharsize is a Sybase global variable that contains the byte length of a character in the Sybase server character set. |
| varchar(n)       | VarChar                        | n      | NA    |
| nvarchar(n)      | VarChar                        | n * @ncharsize | @ncharsize is a Sybase global variable that contains the byte length of a character in the Sybase server character set. |
| sysname          | VarChar                        | 30     | The user data type supplied by Sybase, which is defined as varchar(30). |
| text             | Longvarchar                    | NA     | NA    |

Numeric (integer) data types

The following table summarizes the numeric (integer) data types for Sybase SQL Server and their InfoSphere DataStage SQL type definitions.

Table 5. Numeric (integer) data types

<table>
<thead>
<tr>
<th>Sybase Data Type</th>
<th>InfoSphere DataStage SQL Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit</td>
<td>Bit</td>
</tr>
<tr>
<td>tinyint</td>
<td>Tinyint</td>
</tr>
<tr>
<td>smallint</td>
<td>Smallint</td>
</tr>
<tr>
<td>int</td>
<td>Integer</td>
</tr>
</tbody>
</table>

Numeric (decimal) data types

The following table summarizes the numeric (decimal) data types for Sybase SQL server, their InfoSphere DataStage SQL type definitions, and the corresponding length and scale attributes that you need to set.
Table 6. Numeric (decimal) data types

<table>
<thead>
<tr>
<th>Sybase Data Type</th>
<th>InfoSphere DataStage SQL Type</th>
<th>Length</th>
<th>Scale</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>decimal(p,s)</td>
<td>Decimal</td>
<td>p</td>
<td>s</td>
<td>The full range of Sybase decimal values are supported without loss of precision.</td>
</tr>
<tr>
<td>numeric(p,s)</td>
<td>Numeric</td>
<td>p</td>
<td>s</td>
<td>The full range of Sybase numeric values are supported with no loss of precision.</td>
</tr>
</tbody>
</table>

**Numeric (money) data type**

The following table summarizes the numeric (money) data type for Sybase SQL Server, its InfoSphere DataStage SQL type definitions, and the corresponding length and scale attributes that you need to set.

Table 7. Numeric (money) data types

<table>
<thead>
<tr>
<th>Sybase Data Type</th>
<th>InfoSphere DataStage SQL Type</th>
<th>Length</th>
<th>Scale</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>money</td>
<td>Decimal</td>
<td>19</td>
<td>4</td>
<td>The full range of Sybase money values are supported with no loss of precision.</td>
</tr>
</tbody>
</table>

**Numeric (approximate) data types**

The following table summarizes the numeric (approximate) data types for Sybase SQL server, their InfoSphere DataStage SQL type definitions, and the corresponding length attributes that you need to set.

Table 8. Numeric (approximate) data types

<table>
<thead>
<tr>
<th>Sybase Data Type</th>
<th>InfoSphere DataStage SQL Type</th>
<th>Length</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>float(p)</td>
<td>Float</td>
<td>p</td>
<td>InfoSphere DataStage Float values have a maximum precision of 15 digits. Some loss of precision will occur when reading data from Sybase float(p) columns where p is greater than 15.</td>
</tr>
<tr>
<td>real</td>
<td>Real</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>double precision</td>
<td>Double</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
### Date data types

The following table summarizes the date data types for Sybase SQL Server and their InfoSphere DataStage SQL type definitions.

*Table 9. Date data types*

<table>
<thead>
<tr>
<th>Sybase Data Type</th>
<th>InfoSphere DataStage SQL Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>datetime</code></td>
<td>TIMESTAMP</td>
<td>4 The time component of a Sybase <code>datetime</code> value is lost when converted to a InfoSphere DataStage Date value. When writing a InfoSphere DataStage Date value to a Sybase <code>datetime</code>, the time component is set to midnight.</td>
</tr>
<tr>
<td></td>
<td>Date</td>
<td>1 The date component of a Sybase <code>datetime</code> value is lost when converted to a InfoSphere DataStage Time value. When writing a InfoSphere DataStage Time value to a Sybase <code>datetime</code>, the date component is set to the current date on the InfoSphere DataStage server machine.</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td></td>
</tr>
</tbody>
</table>

| `smalldatetime`  | TIMESTAMP                     | 4 The time component of a Sybase `smalldatetime` value is lost when converted to a InfoSphere DataStage Date value. When writing a InfoSphere DataStage Date value to a Sybase `smalldatetime`, the time component is set to midnight. |
|                  | Date                          | 1 The date component of a Sybase `smalldatetime` value is lost when converted to a InfoSphere DataStage Time value. When writing a InfoSphere DataStage Time value to a Sybase `smalldatetime`, the date component is set to the current date on the InfoSphere DataStage server machine. |
|                  | Time                          |  |

### Binary data types

The following table summarizes the binary data types for Sybase SQL server and their InfoSphere DataStage SQL type definitions.
### Table 10. Binary data types

<table>
<thead>
<tr>
<th>Sybase Data Type</th>
<th>InfoSphere DataStage SQL Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary</td>
<td>Binary</td>
<td>NA</td>
</tr>
<tr>
<td>varbinary</td>
<td>Varbinary</td>
<td>NA</td>
</tr>
<tr>
<td>image</td>
<td>Long Varbinary</td>
<td>NA</td>
</tr>
<tr>
<td>timestamp</td>
<td>Unsupported</td>
<td>The user data type supplied by Sybase, defined as <code>varbinary(8)</code>. <code>timestamp</code> is not compatible with the InfoSphere DataStage Timestamp data type.</td>
</tr>
</tbody>
</table>

### Stored procedure support

You can call stored procedures from the server Sybase OC stage. The following restrictions apply:

- Specify input parameters as literal values. Passing row values as parameter values is not supported.
- Output parameters are not supported.
- You can call stored procedures as part of the BeforeSQL and AfterSQL statements. Any result sets generated by the procedure are discarded.
- You can also call stored procedures as part of the "User Defined SQL Statement" for output and reference links only. The stored procedure must generate a row result set that matches the stage output column definitions. Only one row result set is processed; any additional result sets are discarded.
- To call a stored procedure that is part of a group, precede the semicolon (;) that separates the group name and procedure number with a backslash (\). The backslash causes the stage to treat the semicolon as a regular character instead of an end-of-batch signal. For example, to call the myprocgroup;2 stored procedure from the stage, use the following syntax:
  ```
  execute myprocgroup\;2
  ```
- You cannot call stored procedures for input links. Passing row values as parameter values is not supported.
- The return code for the procedure is checked for Sybase errors (-1 to -99).
- Error numbers from -1 to -8 are treated as nonfatal warnings. Error numbers from -9 to -99 are treated as fatal errors. For more information on stored procedure return codes, see your Sybase documentation.
- You must set the procedure's execute mode to "chained" or "anymode", as Sybase OC executes in chained transaction mode. To set execute mode, use the Sybase `sp_procxmode` procedure. For example:
  ```
  sp_procxmode myproc, "anymode"
  ```

For more information on `sp_procxmode`, see the Sybase documentation.
Chapter 4. BCPLoad stages

The BCPLoad stage uses the BCP (Bulk Copy Program) utility to bulk load data into a single table in a Microsoft SQL server 2000 or Sybase (System 11.5 or 12.5) database.

The BCPLoad stage is a passive connectivity stage. The stage is installed automatically when you install the IBM InfoSphere DataStage and appears in the Database category on the server job palette.

This topic describes the following topics for the BCPLoad stage:

- “BCPLoad stage overview”
- “Table definitions” on page 26
- “SQL data types” on page 26
- “Using BCPLoad stages” on page 26
- “Defining BCPLoad input data” on page 31

Note: Because SQL server is not supported on UNIX operating systems, you cannot use the BCPLoad stage to load data to a SQL server database when your engine tier is on a UNIX system.

BCPLoad stage overview

The BCPLoad stage uses the same API that BCP does, but loads data directly, without the need for a format file. The command line switches are set by using the stage properties.

By default, the BCPLoad stage is configured to bulk load data into Microsoft SQL server. You can configure the BCPLoad stage properties to bulk load data into a Sybase SQL server table by using the Sybase DBLIB or CTLIB client libraries.

Note: The client libraries used by the BCPLoad stage are not supplied as part of the IBM InfoSphere DataStage. You must obtain these libraries from your DBMS vendor, and ensure they are installed and configured on your system, before attempting to use the BCPLoad stage.

Because this version of the BCPLoad stage supports both Microsoft SQL server and Sybase, only BCP switches common to both servers have been included as stage properties. The following command line switches are not supported for Microsoft SQL server:

- -T, trusted connection
- -q, quote identifiers

The following command line switches are not supported for Sybase:

- -L, interface file
- -J, the client character set
- -Q, the data character set

For more information about the BCP switches that can be set, see “Editing stage properties” on page 27.
The BCPLoad stage does not support the loading of native data files.

**Table definitions**

You can import a table definition from the table in your SQL server database by selecting **Import > Table Definitions** from the Designer menu. The table definition is imported via an ODBC connection to the Server. You can then load this table definition into the stage by clicking **Load** on the **Columns** tab on the BCPLoad stage Inputs page.

**SQL data types**

The following SQL server data types are supported by the BCPLoad stage:
- Bit
- Char
- DateTime
- Decimal
- Float
- Integer
- Money
- Numeric
- Real
- SmallDateTime
- SmallInt
- SmallMoney
- TinyInt
- VarChar

When you import metadata from your database table, these data types are mapped to appropriate SQL data types by the ODBC driver. You can view the data types used in the table definition from the repository, or when you edit a stage in your job design.

The following SQL Server data types are not supported by the BCPLoad stage:
- Binary
- VarBinary
- Image
- Text (large text which is a binary type)

**Using BCPLoad stages**

The BCPLoad stage is a target stage. It has one input link, which provides a sequence of rows to load into the SQL Server or Sybase database table. The metadata for each input column determines how it is loaded. There are no output links from this stage.

When you edit the BCPLoad stage, the **BCPLoad Stage** dialog box appears. This dialog box has two pages:
- **Stage**. Contains the name of the stage you are editing. This page has up to three tabs:
– **General.** Contains an optional description of the stage and the stage type (BCPLoad).
– **Properties.** Contains the stage properties and their current values. You can edit the default settings for the stage properties or specify job parameters. For details, see “Editing stage properties.”
– **NLS.** If NLS is enabled and you do not want to use the project default character set map, you can select an alternative character set map from this tab.

- **Inputs.** Specifies the column definitions for the data input link.

Click OK to close this dialog box. Changes are saved when you save the job.

**Must do's for Sybase BCPLoad stages**

**About this task**

This section specifies the minimum steps needed to get a BCPLoad stage functioning.

To use the BCPLoad stage:

**Procedure**

1. Start the Designer and open your server job design.
2. Click the **Sybase BCP Load** button on the tool palette.
3. Click in the Diagram window where you want to position the stage.
4. Link an output from a relevant stage in the job design to the input of the BCPLoad stage.
5. Configure the BCPLoad stage:
   a. Edit stage properties on the **Properties** tab or specify job parameters.
   b. Optionally define a character set map on the **NLS** tab if NLS is enabled.
   c. Define the data on the input links.
   These steps are performed in the BCPLoad Stage dialog box.

**Editing stage properties**

The **Properties** tab on the Stage page allows you to view and edit properties for the BCPLoad stage. It contains a grid displaying the following property names and values:

- **SQL-Server Name.** The name of the SQL server to connect to. This property corresponds to the BCP -S switch. This property is optional and has no default value. If you leave this property blank, the stage assumes the SQL server resides on the same machine as the IBM InfoSphere DataStage Server.
- **User ID.** The logon name of the SQL user. This property corresponds to the BCP -U option. This property is required and there is no default value.
- **Password.** The password of the SQL user. This property corresponds to the BCP -P option. This property is required and there is no default value.
- **Database Name.** The name of the database to use on the SQL Server. This property is required and there is no default value.
- **Table Name.** The name of the table to load data into. This property is required and there is no default value.
• **Before Load Stored Procedure.** The name of a stored procedure that is executed before the database table is loaded. This property is optional and has no default value. For more information about using a before-load stored procedure, see "Using stored procedures."

• **After Load Stored Procedure.** The name of a stored procedure that is executed after the database table is loaded. This property is optional and has no default value. For more information about using an after-load stored procedure, see "Using stored procedures."

• **Batch Size.** The number of rows to include in the BCP batch. This property corresponds to the BCP -b option. The default setting for this property is 0, that is, all the data rows are treated in one batch. If an error occurs, all rows are rolled back.

• **Packet Size.** The number of bytes per network packet sent to and from the server. The default value is 4096. You can specify any number from 512 through 65535.

• **Use Source Identity Data.** This property corresponds to the BCP -E switch. Setting this property tells the SQL Server to use the identity values passed to it by the BCPLoad stage to populate the corresponding identity column in the SQL Server table.

• **Date Format.** This property provides a workaround to the problem that Microsoft SQL Server has with dates in YMD format. If your target table has a date column and your data has dates in YMD format, a conversion is required for the date to load successfully. By setting this property to ymd, dates are automatically converted during loading to a format that Microsoft SQL Server accepts.

• **Client Library.** The type of client library to use. The default setting is MSDBLIB (the Microsoft DBLibrary). Other valid settings are SYBDBLIB for the Sybase DBLibrary and SYBCTLIB for the Sybase CTLibrary. You cannot use the MSDBLIB (the Microsoft DBLibrary) if your engine tier is on a UNIX system.

There are also four buttons on this tab:

• **Insert Job Parameter...** Allows you to insert a job parameter as the value for a selected property. You can use job parameters for any of the properties on this tab. When you validate or run the job, you are prompted to specify suitable values for the properties.

  When you click this button, a list appears displaying the currently defined job parameters. Choose a parameter from the list or click (New...) to define a new one. The Job Properties dialog box appears with the Parameters tab displayed. You can also insert a job parameter by using the F9 key.

• **Set to Default.** Sets the value for the selected property to the default value.

• **All to Default.** Sets the values for all properties to the default values.

• **Property Help.** Displays the help text supplied by the creator of the stage definition.

You can edit the value for any property listed in the grid. Click OK to save the settings and close the BCPLoad Stage dialog box.

**Using stored procedures**

You can specify the name of a stored procedure to run before or after loading the database. Before-load stored procedures can be used to perform tasks such as dropping indexes and turning on the database bulk copy option. After-load stored
procedures can be used to turn off the bulk copy option and recreate any indexes. For a detailed description of how to write a stored procedure, see the SQL Server documentation.

The stored procedure name is entered as the value for the **Before Load Stored Procedure** or **After Load Stored Procedure** stage property. As well as entering the name of a stored procedure, you can also include parameter values. To specify parameters for the stored procedure, use the following format in the Value field on the Properties tab:

```
procedurename P1, P2, P3, ..., Pn
```

*procedurename* is the name of the stored procedure.

*P1...Pn* are parameter values, in the order expected by the stored procedure. Note that string values must be quoted.

If you want to return messages from a stored procedure and write them to the job log file, you can use the output parameters DSSeverity and DSMessage. These parameters return messages to the job log file with an appropriate severity. The type of message written to the job log file depends on the value returned by the DSSeverity parameter:

- Return value of 0. Nothing is written.
- Return value of 1. An informational message is written.
- Return value of 2. A warning message is written.
- Return value of 3. A fatal message is written. The IBM InfoSphere DataStage job aborts and any return values from the stored procedure, other than the InfoSphere DataStageWebSphere expected output parameters, are ignored.

The following example is of a before-load stored procedure. This stored procedure demonstrates the use of the output parameters DSSeverity and DSMessage:

```sql
create proc DemoBeforeSP
    @lReplace bit,
    @DSSeverity int output,
    @DSMessage varchar(255) = "" output
as
/* Remove the following three lines if running on Sybase */
declare @sSetDBOption varchar(255)
select @sSetDBOption = 'sp_dboption' + DB_NAME() + ', ' + 'select 
into/bulkcopy', TRUE"
exec (@sSetDBOption)
if @lReplace = 1
    begin
        truncate table BCPLoadSample
    end
if @ERROR = 0
begin
    select @DSMessage = "Before SP completed: "
    if @lReplace = 1
        begin
            select @DSMessage = @DSMessage + "replacing existing data"
        end
    else
        begin
            select @DSMessage = @DSMessage + "appending data"
        end
    select @DSSeverity = 1 /* INFO */
end
else
begin
```
To use this stored procedure, type DemoBeforeSP 1,DSSeverity, DSMessage as the value for the Before Load Stored Procedure property when you edit stage properties.

To use existing stored procedures, type the name of the stored procedure and appropriate parameter values as the value for the Before Load Stored Procedure or After Load Stored Procedure property.

For example, suppose your stored procedure includes this:

```sql
create proc sp_TrustyDebuggedProcedure
    @strTableName char(30),
    @strSurname char(30),
    @iRowCount int = 0 output
as
...
...
```

If you want to use this procedure as a before-load procedure, you would type `sp_TrustyDebuggedProcedure "Table1","Smith"` in the Value field for the Before Load Stored Procedure property. "Table1" and "Smith" are passed in as `strTableName` and `strSurname` respectively.

If you want to modify an existing stored procedure to return a severity warning and an error message, the create procedure needs to be modified to include the two output parameters DSSeverity and DSMessage. In the earlier example, the create procedure would become:

```sql
create proc sp_TrustyDebuggedProcedure
    @strTableName char(30),
    @strSurname char(30),
    @iRowCount int = 0 output,
    @DSSeverity int output,
    @DSMessage varchar(255) = "" output
as
...
.../* Somewhere in the procedure set appropriate values for*/
.../* DSSeverity and DSMessage*/
```

In this case, you would type the following text in the Value field for the Before Load Stored Procedure:

`sp_TrustyDebuggedProcedure "Table1","Smith",0,DSSeverity,DSMessage`

You can include job parameters to represent the value of a stored procedure parameter. To use job parameters in the earlier example, you would type the following text in the Value field for the Before Load Stored Procedure:

`sp_TrustyDebuggedProcedure #Table#,#Name#,0,DSSeverity,DSMessage`

`Table` and `Name` are the names of two defined job parameters.

### Defining character set maps

You can define a character set map for the BCPLoad stage by using the NLS tab on the Stage page. You can choose a specific character set map from the list, or accept the default setting for the whole project. This tab also has the following options:
• **Show all maps.** Displays all the maps supplied with the IBM InfoSphere DataStage. Maps cannot be used unless they have been loaded by using the IBM InfoSphere DataStage and QualityStage Administrator.

• **Loaded maps only.** Displays the maps that are loaded and ready for use.

• **Use Job Parameter...** Allows you to specify a character set map as a parameter to the job containing the stage. If the parameter has not yet been defined, you are prompted to define it from the Job Properties dialog box.

### Defining BCPLoad input data

When you write data to a file in BCP load format, the BCPLoad stage has an input link. The properties of this link and the column definitions of the data are described on the Inputs page in the BCPLoad Stage dialog box. This page has two tabs:

• **General.** Contains an optional description of the link.

• **Columns.** Contains the column definitions for the data you are loading into your database table. These column definitions are usually specified by the metadata defined on the output link of the previous stage in the job design. If the columns are not already defined, you can click **Load** to load columns from a table definition in the repository, or you can type column definitions manually and click **Save...** to save them as a table definition.
Chapter 5. Sybase enterprise stage

The Sybase enterprise stage is a database stage that allows you to read data from and write data to a Sybase database. You can also use the Sybase enterprise stage in conjunction with a Lookup stage to access a lookup output table hosted by a Sybase database. The Sybase enterprise stage can have one input link and one reject link, or one output link or output reference link. The stage operates in the modes listed below:

- Read
- Write
- Upsert
- Lookup

Note: To perform read and write operations on a table hosted by a Sybase IQ server, make sure that the Sybase IQ server is running local to the IBM InfoSphere DataStage Server. To perform such operations on a table hosted by a Sybase ASE server, the server can be on any system, whether local or remote to the InfoSphere DataStage.

Prerequisites

Sybase Open client or SDK 12.5.1, Version 19 or later is a prerequisite for using Sybase Enterprise Edition stage on Microsoft Windows. For the Sybase Enterprise stage to work correctly, install or upgrade to Sybase Open client 12.5.1, Version 19 or later. To load data to remote the Sybase IQ server, make sure that Sybase Open client 15.5 and Sybase IQ network client 15.1 ESD 1 or higher are installed on the engine tier of the InfoSphere Information Server.

Sybase enterprise stages

Read
Use the Sybase enterprise stage in Read mode to read a Sybase table.

Write
Use the Sybase enterprise stage in Write mode to load data to a table hosted by a Sybase IQ or Sybase ASE Server.

Upsert
Sybase Enterprise stage uses the Sybase Enterprise stage in Upsert mode to update, insert, and delete data from a table hosted by a Sybase IQ or Sybase ASE server.

Lookup
Use the Sybase enterprise stage to perform a lookup operation directly on a Sybase table or load a Sybase table into memory and then perform a lookup operation on the table.

When you use a Sybase stage as a source for lookup data, there are special considerations about column naming. If you have columns of the same name in both the source (table) and lookup data sets, only the source data set column goes to the output data. If you want this column to be replaced by the column from the lookup data source, you must drop the source data column before you perform the lookup.
Environment variables

The following environment variables are available for Sybase Enterprise Edition stage.

Environment variables that affect Sybase IQ read operation

**APT_SYBASE_NULL_AS_EMPTY**
Set this environment variable to 1, to extract null values as empty string. This environment variable has no effect when fixed width extract is done.

**APT_IMPEXP_ALLOW_ZERO_LENGTH_FIXED_NULL**
Prerequisite for APT_SYBASE_NULL_AS_EMPTY.

Environment variables that affect Sybase IQ write operation

**APT_SYBASE_NULL_AS_EMPTY**
Set this environment variable to 1, to load "" as null value. To use this functionality, Load_ZeroLength_AsNull option must be supported and enabled.

**APT_IMPEXP_ALLOW_ZERO_LENGTH_FIXED_NULL**
Prerequisite for APT_SYBASE_NULL_ASEMPTY.

**APT_SYBASE_PRESERVE_BLANKS**
Set this environment variable to load blanks to varchar.

**APT_SYBASE_CONVERSION_ERROR**
Set this environment variable to turn off conversion error warnings during load.

**APT_SYBASE_CHECKPOINT_OFF**
Set this environment variable to disable checkpoint error warnings during load.

**APT_SYBASE_LOAD_MEMORY_MB**
Use this environment variable to set the load memory.

Loading data to remote Sybase IQ server

The Sybase Enterprise stage supports loading data to remote Sybase IQ server.

**Before you begin**

Make sure that the prerequisites are met.

**Procedure**

1. Connect to the Sybase IQ 15.0 server.
2. Select `allow_read_client_file, allow_write_client_file` and change the value to On. The default value is Off.
3. To set the server option property `allow_read_client_file` to On. Run the following commands using isql or dbisql utility:
   ```sql
   set option [allow_read_client_file] = on 
   grant readclientfile to <group|user>
   ```
4. Create the environmental variable APT_SYBASE_REMOTE_LOAD and set it to 1.
Chapter 6. Sybase enterprise stage editor

To edit a Sybase enterprise stage, use the Sybase enterprise stage editor. The Sybase enterprise stage editor is based on the generic stage editor.

The stage editor has as many as three pages, depending on whether you are reading a database or writing to a database:

- Stage page
- Input page
- Output page

### Stage page

This page is always present. You use this page to specify general information about the Sybase enterprise stage. The following tabs are present on this page:

- **General**: Use the General tab to specify a description of the Sybase enterprise stage. This is an optional feature.
- **Advanced**: The Advanced tab allows you to specify how the stage executes. Use this tab to specify the following:
  - **Execution Mode** - The stage can execute in parallel mode or sequential mode. In parallel mode, data is processed by the available nodes as specified in the Configuration file, and by any node constraints specified by using the options available under the Advanced tab. The conductor node processes the data in sequential mode. Sybase enterprise stage runs in sequential mode.
  - **Combinability mode** - This property is set to Auto by default. This allows the IBM InfoSphere DataStage to combine the stages that underlie parallel stages so that they run in the same process if allowed.
  - **Preserve partitioning** - Select Set or Clear. If you select Set, read operations will request that the next stage preserves the partitioning as is. This field is visible only if the stage has output links.
  - **Node pool and resource constraints**. Select this option to constrain parallel execution to one or more node pools or resource pools specified in the grid. The grid allows you to make choices from the lists that are populated from the Configuration file.
  - **Node map constraint**. Select this option to constrain parallel execution to the nodes in a defined node map. You can define a node map by typing node numbers into the field. Alternatively, click the browse [...] that appears on the right side of the field to open the Available Nodes dialog box and select appropriate nodes. You are effectively defining a new node pool for this stage, in addition to any node pools defined in the Configuration file.

- **NLS Map**: The NLS Map tab appears if you have NLS enabled on your system. Use the NLS Map tab to define a character set map for the Sybase enterprise stage. By defining a character set map, you override the default character set map set for a project or a job. You can specify that the map be supplied as a job parameter if required.
The stage editor includes this page when you are writing data to a Sybase database. Use the Input page to specify details about how the Sybase enterprise stage writes data to a Sybase database. The Sybase enterprise stage can have only one input link writing to one table. The tabs on the Input page that you will mainly use to set up Sybase enterprise stage are:

- General
- Properties
- Columns
- Advanced

The use of each of these tabs is described below.

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

**Properties tab on the input page**
Use the Properties tab to specify properties for the input link. The properties dictate how incoming data is written and where. Some of the properties are mandatory, and many have default settings. Properties without default settings appear in the warning color (red by default) and turn black when you supply a value for them. The following table gives a quick reference list of the properties and their attributes. A more detailed description of each property follows.

**Table 11. Target properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
<th>Default</th>
<th>Required?</th>
<th>Dependent on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>String</td>
<td>N/A</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Write Method</td>
<td>• IQ</td>
<td>IQ Write</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Write Mode</td>
<td>• Append</td>
<td>Append</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Auto-generated</td>
<td>Yes/No</td>
<td>Yes</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Upset SQL</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Insert SQL</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Insert Array Size</td>
<td>Number</td>
<td>1</td>
<td>N</td>
<td>Insert SQL</td>
</tr>
<tr>
<td>Update SQL</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Delete SQL</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Write Method</td>
<td>• Upset</td>
<td>Write</td>
<td>Y</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Table 11. Target properties (continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
<th>Default</th>
<th>Required?</th>
<th>Dependent on</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASE Upsert or IQ Upsert Mode</td>
<td>Insert only</td>
<td>Insert then update</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Delete only</td>
<td>Delete then Insert</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Insert only</td>
<td>Insert then update</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update only</td>
<td>Update then Insert</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Insert then update</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 12. Connection properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
<th>Default</th>
<th>Required?</th>
<th>Dependent on</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>String</td>
<td>N/A</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Password</td>
<td>String</td>
<td>N/A</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Server</td>
<td>String</td>
<td>N/A</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Database</td>
<td>String</td>
<td>N/A</td>
<td>Y</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 13. Options properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
<th>Default</th>
<th>Required?</th>
<th>Dependent on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate Column Name</td>
<td>True/False</td>
<td>False</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Default String Length</td>
<td>Number</td>
<td>1</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Open Command</td>
<td>String</td>
<td>N</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Close Command</td>
<td>String</td>
<td>N</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Length to Truncate</td>
<td>Number</td>
<td>N</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Drop Unmatched Field</td>
<td>True/False</td>
<td>False</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Row Commit Interval</td>
<td>Number</td>
<td>1</td>
<td>N</td>
<td>Insert Array Size</td>
</tr>
<tr>
<td>Max Reject Records</td>
<td>Number</td>
<td>N/A</td>
<td>N</td>
<td>Options/ASE Write Output Reject Records</td>
</tr>
<tr>
<td>Output Reject Records</td>
<td>True/False</td>
<td>False</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Use Identity</td>
<td>True</td>
<td>False</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>BatchSize</td>
<td>Integer</td>
<td>1000</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Fixed Width</td>
<td>True/False</td>
<td>False</td>
<td>N</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Setting the Sybase enterprise stage properties for the input link

About this task

This section describes how to use the fields and options available under the Properties tab of the input link to set up the stage properties appropriately.

- Under the Target category, set the following properties:
  - **Table** - Specify the name of the table to which you want to write data. You can specify this value as a job parameter if required.
  - **Write Method** - Select IQ Upsert, IQ Write, ASE Upsert, or ASE Write. IQWrite is the default value. If you select IQUpsert or ASEUpsert, then you must specify the Insert SQL and Update SQL statements. Select IQWrite or ASEWrite to set up a connection to a Sybase database and then insert records from one input dataset into a table. The Sybase IQ Upsert operations are expected to insert/update the data in a local or remote Sybase IQ database. The connection parameters can point to a remote or local IQ database.
  - **Write Mode** - Select the appropriate write mode to determine how the records of a dataset are inserted into the table. The available options are:
    - **Append:** This is the default write mode. Select this mode to append new records to an existing table.
    - **Create:** Select this mode to create a new table. If the table already exists, an error occurs and the job terminates. You must specify this mode if the destination table does not exist.
    - **Replace:** Select this mode to first drop the existing table and then create a new table in its place.
    - **Truncate:** Select this mode to retain the attributes of the existing table, including schema, discard existing records, and then append new records to the table.
  - **Auto-generate SQL** - This property appears only if you select IQ Upsert or ASE Upsert as the write method (see the Write Method section above). The table below describes how to setup this property:

<table>
<thead>
<tr>
<th>Option</th>
<th>If Auto-generated Upsert SQL = Yes</th>
<th>If Auto-generated Upsert SQL = No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete only</td>
<td>Select this option to have a delete statement generated automatically, based on the table and column details that you provide. When you select Delete Only as the Upsert mode, an additional property Delete SQL appears on the list. You must select an appropriate value for the Delete SQL property. To view the automatically generated statement, click Delete SQL. The statement appears in a field on the right side of the list of properties.</td>
<td>Select this option to create your own delete statement. Click Delete SQL, and then enter the statement in the Delete SQL field on the right side of the list of properties.</td>
</tr>
<tr>
<td>Delete then Insert</td>
<td>Select this option to have delete and insert statements generated automatically, based on the table and column details that you provide. When you select Delete then Insert as the Upsert mode, two additional properties Delete SQL and Insert SQL appear on the list. You must select appropriate values for these two properties. To view the automatically generated statements, click Delete SQL and Insert SQL. The statements appear in their respective fields on the right side of the list of properties.</td>
<td>Select this option to create your own delete and insert statements. Click Delete SQL and then enter the statement in the Delete SQL field on the right side of the list of properties. Next, click Insert SQL and then enter the statement in the Insert SQL field on the right side of the list of properties.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Insert only</td>
<td>Select this option to have an insert statement generated automatically, based on the table and column details that you provide. When you select Insert only as the Upsert mode, an additional property Insert SQL appears on the list. You must select an appropriate value for this property. The statement appears in a field on the right side of the list of properties.</td>
<td>Select this option to create your own insert statement. Click Insert SQL, and then enter the statement in the Insert SQL field on the right side of the list of properties.</td>
</tr>
<tr>
<td>Insert then Update</td>
<td>Select this option to have insert and update statements generated automatically, based on the table and column details that you provide. When you select Insert then Update as the Upsert mode, two additional properties Insert SQL and Update SQL appear on the list. You must select appropriate values for these two properties. The statements appear in their respective fields on the right side of the list of properties.</td>
<td>Select this option to create your own insert and update statements. Click Insert SQL and then enter the statement in the Insert SQL field on the right side of the list of properties. Next, click Update SQL and then enter the statement in the Update SQL field on the right side of the list of properties.</td>
</tr>
</tbody>
</table>
Table 14. Options for Auto-generate SQL property (continued)

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Update only</strong></td>
<td>Select this option to have an <em>update</em> statement generated automatically, based on the table and column details that you provide. The <em>Update SQL</em> property appears on the list by default. You must select an appropriate value for this property. The statement appears in a field on the right side of the list of properties.</td>
</tr>
<tr>
<td><strong>Update then Insert</strong></td>
<td>Select this option to have <em>update and insert</em> statements generated automatically, based on the table and column details that you provide. When you select <em>Update then Insert</em> as the <em>Upsert mode</em>, two additional properties <em>Update SQL</em> and <em>Insert SQL</em> appear on the list. You must select appropriate values for these two properties. The statements appear in their respective fields on the right side of the list of properties.</td>
</tr>
</tbody>
</table>

Select this option to create your own *update* statement. Click *Insert SQL*, and then enter the statement in the *Insert SQL* field on the right side of the list of properties.

Select this option to create your own *update and insert* statements. Click *Update SQL* and then enter the statement in the *Update SQL* field on the right side of the list of properties. Next, click *Insert SQL* and then enter the statement in the *Insert SQL* field on the right side of the list of properties.

**Note:** *Insert SQL* has a dependent property *Insert Array Size*. You must specify the size of the insert host array for *Insert Array Size*. The default size is 1 record. If you want each *Insert SQL* statement to be executed individually, specify 1 for this property.

- **Under the Connection category**, specify values for the following:
  - **User**: Enter the user name. This is a required property.
  - **Password**: Enter the password. This is a required property.
  - **Server**: Specify the server name to be used for all Sybase database connections. This is a required property.
  - **Database**: Enter the database name. This is optional, and if not specified, connects to the default database.

Specify the above by using any one of the methods below:

- **Method 1**: Enter the value in the corresponding field that appears on the right side of the *Properties* list.
- **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 popup appears with a list of available job parameters from which you can choose. If you wish to create a new parameter for the job, click [New...] from the popup list, and create an appropriate environment variable by using the Job Properties dialog that appears.

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

- **Under the Options category**, specify values for the following:
- **Truncate Column Names** - This only appears for the Write Method of IQ Write or ASE Write. You can set the value as True or False. Select True to truncate field names to the size allowed by the Sybase. Select False to disable truncation of field names.

- **Default String Length**: This is an optional property and appears only for the Write Method of IQ Write or ASE Write. It is set to the size of 1 byte by default. By specifying a value for this property, you set the default string length of variable-length strings written to a table.

  The maximum length you can set is 2000 bytes. Note that the stage always allocates the specified number of bytes for a variable-length string. In this case, setting a value of 2000 allocates 2000 bytes for every string. Therefore, you should set the expected maximum length of your largest string and no larger. In case of NLS maximum = 2000 / number of bytes per NLS character. For example, in a Unicode database, a Japanese character occupies 3 bytes.

- **Open Command**: Use this property to specify a command in single quotes. This command is to be parsed and executed by the Sybase database on all processing nodes before the table is opened. You can enter the command directly or specify the value for this property as a job parameter.

- **Close Command**: Use this property to specify a command in single quotes. This command is to be parsed and executed by the Sybase database on all processing nodes after the stage finishes processing the table. You can enter the command directly or specify the value for this property as a job parameter.

- **Create Statement**: This option appears for a Sybase ASE write operation only if you select a write mode of Create or Replace, under the Target category. Use this option to create the table that you want to create or replace.

- **Length to Truncate**: Enter the length to which to truncate columns names.

- **Drop Unmatched Field**: You can set this property as True or False. Set True to silently drop all input columns that do not correspond to columns in an existing table. Otherwise the stage reports an error and terminates the job. The default value is False.

- **Row Commit Interval**: Specify the number of records to be committed before the start of a new transaction.

- **Output Reject Records**: This property appears only for the Write Method of IQ Upsert, IQ Write, or ASE Write. The default value is False. Select True to send rejected records to the reject link.

- **Max Reject Records**: This option appears if you set the Output Reject Records option to True. You can specify a maximum number of records to be sent to the reject link.

- **Use Identity**: This property is available when the Write Method is set to ASE Write. Use Identity controls the value of identity columns. When the property is set to True, the values for identity columns are generated by the server. When the property is set to false, the values are inserted from incoming data.

- **BatchSize**: Set this option to specify the transaction size for bcp load. This option is available only when write method is set to ASE.

- **Fixedwidth**: Set this option to load rows as fixed width ASCII. This option is available when write method is set to IQ. This option increases the performance of load option.

**Note**: Sybase IQ does not support partitioning. Therefore, you can ignore the Partitioning tab, and let the Sybase enterprise stage take the default values.
Columns tab on the input page
Use the Columns tab to specify the column definitions for the incoming data.

Advanced tab on the input page
Use the Advanced tab to modify the default buffering settings for the input link.

Output page
The Sybase enterprise stage editor includes this page when you are reading from a Sybase database or performing a lookup on a Sybase database. Use the Output page to specify details about how the Sybase enterprise stage reads data from a Sybase database. The Sybase enterprise stage can have only one output link. Alternatively, it can have a reference output link. The lookup stage uses this reference output link when referring to a Sybase lookup table. It can also have a reject link to which rejected records are routed. This reject link is used in conjunction with an input link. Use the Output Name list to choose to view details about the main output link or the reject link.

The Output page has the following tabs:
• General
• Properties
• Columns
• Advanced

The below sections describe how to use each of these tabs.

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

Properties tab on the output page
Use the Properties tab to specify the properties of the output link. The properties that you specify dictate how incoming data is read from which table. Many of these properties have default settings. Properties without default settings appear in the warning color (red by default) and turn black when you supply a value for them.

The following table is a quick reference list of the properties and their attributes. A more detailed description of each property follows the table.

Table 15. Source properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
<th>Default</th>
<th>Required?</th>
<th>Dependent on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Type</td>
<td>• Normal</td>
<td>Normal</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>• Sparse IQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Sparse ASE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read DB Type</td>
<td>IQ/ASE</td>
<td>IQ</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Read Method</td>
<td>Table/User-defined SQL</td>
<td>Table</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Query</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Table</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Selectlist</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>Table</td>
</tr>
<tr>
<td>Where</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>Table</td>
</tr>
</tbody>
</table>
Table 16. Connection properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
<th>Default</th>
<th>Required?</th>
<th>Dependent on</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>String</td>
<td>N/A</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Password</td>
<td>String</td>
<td>N/A</td>
<td>Y</td>
<td>N/A</td>
</tr>
<tr>
<td>Server</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Database</td>
<td>String</td>
<td>N/A</td>
<td>N</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 17. Options properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Values</th>
<th>Default</th>
<th>Required?</th>
<th>Dependent on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Command</td>
<td>String</td>
<td>N</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Close Command</td>
<td>String</td>
<td>N</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Fetch Array Size</td>
<td>Number</td>
<td>1</td>
<td>N</td>
<td>N/A</td>
</tr>
<tr>
<td>Fixed Width</td>
<td>True/False</td>
<td>False</td>
<td>N</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Setting the Sybase enterprise stage properties for the output link

About this task

This section describes how to use the fields and options available under the Properties tab of the output link to set up the stage properties appropriately.

- **Under the Source category**, specify the following values:
  - **Lookup Type**: Sybase enterprise stage is connected to a lookup stage via a reference link. You can perform a normal or in-memory lookup on a Sybase database. To perform an in-memory lookup, set **Lookup Type** to **Normal**. If the lookup type is **Normal**, the **Lookup** stage can have multiple reference links. To perform a direct lookup, set **Lookup Type** to **Sparse IQ** or **Sparse ASE** as required. If the lookup type is **Sparse**, the **Lookup** stage can have one reference link.
  - **Read DB Type**: Select the name of the database type from which data has to be read.

  **Note**: When you are accessing a table from a remote Sybase IQ server, you need to choose the ASE option. This is a short workaround to enable remote reads. Also remote reads are slower than local reads.
  - **Read Method**: Specify a table or a query to perform a read operation on a Sybase database.
  - **Query**: Select this read method to specify an SQL query to read a table. The query specifies the table and the processes that you want to perform on the table. This statement can contain join or view operations, database links, synonyms, and so on. You must specify whether the query is generated automatically or you want to define query yourself. Select **Auto-generated SQL** to have the IBM InfoSphere DataStage automatically generate an SQL query based on the columns you have defined and the table you specify in the Table property. Select **User-defined SQL** to define your own query. A user-defined or auto-generated SQL is read sequentially on one node.

Currently the Sybase enterprise stage does not support complex queries containing Sybase SQL statements such as ifnull and case as a part of the corresponding user-defined queries. However, the stage allows an alternative way to support these queries. You can create a view with the required user-defined SQL and you can mention the view name in the query option, for example, "select * from <view name>".
Table - If you have chosen Table as the read method, then you must specify the name of the source Sybase table. The table must exist and you must have SELECT privileges on the table.

Before you select a value for this option, you must fulfil the below dependencies:

- Use the WHERE clause in your SELECT statement to specify the rows of the table to be included or excluded from the read operation. If you do not supply a WHERE clause, all rows are read.
- You can specify in your SELECT statement the columns that you wish to read. You must specify the columns in this list in the same order as they are defined in the record schema of the input table.

Under the Connection category, set the following properties:

- **User**: Enter the user name. This property is mandatory.
- **Password**: Enter the password. This property is mandatory.
- **Server**: Specify the server name to be used for all Sybase database connections. This property is mandatory.
- **Database**: Enter the database. This is optional and if not specified connects to the default database.

Specify the above by using any one of the methods below:

- **Method 1**: Enter the value in the corresponding field that appears on the right side of the Properties list.
- **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 popup appears with a list of available job parameters from which you can choose. If you wish to create a new parameter for the job, click [New...] from the popup list, and create an appropriate environment variable by using the Job Properties dialog that appears.

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

Under the Options category, set up the following properties:

- **Close Command**: Specify a command to be parsed and executed by the Sybase database on all processing nodes after the stage finishes processing the Sybase table. You can enter a value directly or insert it as a job parameter.
- **Open Command**: Use it to specify any command to be parsed and executed by the Sybase database on all processing nodes before the Sybase table is opened. You can enter a value directly or insert it as a job parameter.
- **BatchSize**: Set this option to specify the transaction size for bcp load. This option is available only when write method is set to ASE.
- **Fetch Array Size**: Specify the number of rows to be retrieved during each fetch operation. The default value is 1.
- **Fixedwidth**: Set this option to extract rows as fixed width ASCII. This option is available when read db type is set to IQ.

**Columns tab on the output page**

Use the Columns tab to specify the column definitions for the data that is being processed.

**Advanced tab on the output page**

Use the Advanced tab to modify the default buffering settings for the output link.
Chapter 7. Working with Sybase databases

IBM InfoSphere DataStage has several defaults. Therefore, including Sybase enterprise stages in a job is very easy. This section describes the minimum steps to enable the functioning of a Sybase enterprise stage. These steps depend upon the purpose of using a Sybase enterprise stage. Before you perform the below steps, you must ensure that column metadata has been specified for any operation that you want to carry out, such as read, write, and load.

Using Sybase open client

The Sybase Enterprise Edition stage libraries are compiled against OSC 12.5. For libraries to load and work correctly with OSC 15.0, run the script lnsyblibs.sh in the $SYSBASE/OSCS-15_0/scripts directory.

For Microsoft Windows, the script is copylibs.bat and for UNIX it is lnsyblibs. Run the script before connecting to Sybase ASE 15.0 Server from Sybase ASE 15.0 client. Instructions on how to run the script are provided in the New Features Open Server 15.0 and SDK 15.0 for Windows, Linux, and UNIX documentation on the Sybase Web site.

Updating a Sybase database

About this task

To update a Sybase database, you must setup the properties of the Sybase Enterprise stage appropriately. You must specify appropriate values for the sub-properties of the Target, Connection, and Option properties.

To setup the properties of the stage, double-click the Sybase Enterprise icon. You see fields and properties under the Properties tab by default.

1. Under the Target category, specify the update method as follows:
   - Select IQ Upsert or ASE Upsert as the Write Method.
   - Specify the table to which to write data.
   - For the Auto-generate Upsert SQL option, select Yes if you wish to use a statement automatically generated by the IBM InfoSphere DataStage. Select No if you wish to specify your own SQL statement. If you select No, then you must specify the Insert/Update/Delete SQL statement to use according to the Upsert Mode selection. If you select Yes, you can edit the automatically generated statement as required.
   - Choose the Upsert Mode. Specify whether to insert/update/delete/insert then update/update then insert/delete then insert data.

2. Under the Connection category, specify connection details for server, user, and password. You can enter these values directly or insert the values by using the job parameter popup list. In case you want to create a new job parameter, you must create a new environment variable for that parameter. You can create parameters at the job level or at the project level. By default, the InfoSphere DataStage connects to the Sybase default database. However, you can specify a different database if required.

3. Under the Options category:
• **Output Reject Records**: This appears only for the **Write Method** of IQ Upsert, IQ Write, or ASE Write. It is **False** by default. Set it to **True** to send rejected records to the reject link.

---

**Loading a Sybase database**

**About this task**

To load a Sybase database, you must setup the properties of the Sybase enterprise stage appropriately. You must specify appropriate values for the sub-properties of the **Target**, **Connection**, and **Option** properties. The graphic below is a sample job designed to load a Sybase database.

To setup the properties of the stage, double-click the **Sybase Enterprise** icon. You see fields and properties under the **Properties** tab by default.

1. **Under the Target category:**
   - Specify a **Write Method** of ASE Write or IQ Write.
     - **ASE Write** uses bcp to load data into a table. Bcp can run in fast or slow mode. If any triggers or indexes have been defined on table to write to, bcp automatically runs in slow mode, and you do not have to set any specific database properties. Otherwise, bcp runs in fast mode. However, bcp cannot run in fast mode unless you set the database property **Select into/bulkcopy** to **True**. To set this property, run the following commands by logging in as a system administrator by using the iSQL utility.
       ```
       use master
go
       sp_dboption <database name>, "select into/bulkcopy", true
       go
       use <database name>
go
       checkpoint
go
       ```
   - Specify the table to which you are writing.
   - Specify the **Write Mode**. By default, the IBM InfoSphere DataStage appends to existing tables. You can also choose to create a new table, replace an existing table, or keep existing table details but replace all the rows.

2. **Under the Connection category,** specify connection details for server, user, and password. You can enter these values directly or insert the values by using the job parameter popup list. In case you want to create a new job parameter, you must create a new environment variable for that parameter. You can create parameters at the job level or at the project level. By default, the InfoSphere DataStage connects to the Sybase default database. However, you can specify a different database if required.

3. **Under the Options category,** specify the following properties:
   - **Truncate Column Names**: This only appears for the **Write Method** of IQWrite or ASE Write. You can set the value as True or False. Set True to truncate field names to the size allowed by the Sybase. Set False to disable the truncation of field names.
   - **Default String Length**: This is an optional property and only appears for the **Write Method** of IQWrite or ASE Write. It is set to the size of 1 byte by default. This property sets the default string length of variable-length strings written to a table. The maximum length you can set is 2000 bytes. Note that the stage always allocates the specified number of bytes for a variable-length string. In this case, setting a value of 2000 allocates 2000 bytes for every string. Therefore, you should set the expected maximum length of your...
largest string and no larger. In case of NLS maximum = 2000 / number of bytes per NLS character. For example, in a Unicode database, a Japanese character occupies 3 bytes.

- **Use Identity:** This only appears for the Write Method of ASE Write. This option is used to control the identity column values. When this option is set to True, the values for the identity column are generated by the server. When this option is set to False, the values are generated from the incoming data.

- **Open Command:** Use it to specify a command, in single quotes, to be parsed and executed by the Sybase database on all processing nodes before the table is opened. You can specify a job parameter if required.

- **Close Command:** Use it to specify any command, in single quotes, to be parsed and executed by the Sybase database on all processing nodes after the stage finishes processing the table. You can specify a job parameter if required.

- **Create Statement:** This option appears only for a Sybase ASE write operation. The Create Statement property appears only if you select a write mode of Create or Replace, under the Target category. Use this option to create the table that you want to create or replace.

- **Length to Truncate:** Enter the length to which to truncate columns names.

- **Drop Unmatched Field:** You can set one of two values, either True or False. Set True to silently drop all input columns that do not correspond to columns in an existing table. Otherwise the stage reports an error and terminates the job. The default value is False.

- **Row Commit Interval:** Specify the number of records to be committed before the start of a new transaction. This option is considered only if Insert Array Size is set to 1 or Row Commit Interval is set to Insert Array Size.

- **Output Reject Records:** This appears only for the Write Method of IQ Upsert, IQ Write, or ASE Write. It is False by default. Set it to True to send rejected records to the reject link. To send bad records down the reject link, you must set the environment variable APT_IMPEXP_ALLOW_ZERO_LENGTH_FIXED_NULL.

- **Max Reject Records:** This property appears only for a Sybase ASE write operation. The Max Reject Records property appears if you set the Output Reject Records property to True. You can specify a maximum number of records to be sent to the reject link.

---

### Reading a Sybase database

#### About this task

To read a Sybase database, you must setup the properties of the Sybase enterprise stage appropriately. You must specify appropriate values for the sub-properties of the **Source, Connection, and Option** properties. The graphic below is a sample job designed to read a Sybase database.

To setup the properties of the stage, double-click the Sybase Enterprise icon. You see fields and properties under the **Properties** tab by default.

- **Under the Source category:**
  - Select the **Read DB Type** as IQ or ASE. IQ is the default value for this property.
  - Choose a **Read Method**. The value for this property is Table by default. You can also choose Auto-generated SQL or User-defined SQL. The read operation takes place sequentially on a single node. If you select User-defined
SQL as the Read Method, then you must specify the SELECT SQL statement to use. To define your own SQL statement, you must edit the automatically generated SQL statement.

Note: Sybase ASE and Sybase IQ read operations do not support the use of functions such as trim and isnull in user-generated SQL statements. However, you can still specify your SQL statements for a read operation. To do this, create a temporary view with the SQL statement that you want to use, and perform a select operation from that view. Create a view `<temporary view name>` as `<user-generated SQL statement>`. Then specify the user-generated SQL statement as: "select * from `<temporary view name>`"

- Specify the table to be read.
- Under the Connection category, specify connection details for server, user, and password. You can enter these values directly or insert the values by using the job parameter popup list. In case you want to create a new job parameter, you must create a new environment variable for that parameter. You can create parameters at the job level or at the project level. By default, the IBM InfoSphere DataStage connects to the Sybase default database. However, you can specify a different database if required.

Reading a remote Sybase IQ server

About this task

Sybase enterprise stage does not support any operations on a remote Sybase IQ server, but supports operations on a remote Sybase ASE server. However, you can read data from a remote Sybase IQ server by using the Read DB Type of ASE.

To read data from a remote Sybase IQ server:

- Under the Source category:
  - Select the Read DB Type as ASE.
  - Choose a Read Method. This is Table by default, but you can also choose to read by using the auto-generated SQL or user-generated SQL. The read operates sequentially on a single node.
  - Specify the table to be read.
  - If you are using a Read Method of user-generated SQL, specify the SELECT SQL statement to use. InfoSphere DataStage provides the editable auto-generated statement.
- Under the Connection category, specify the name of the Sybase IQ server to read, and the user name and password to access the Sybase IQ server.
- Under the Options category, specify values for the following options:
  - Open Command: Use it to specify a command, in single quotes, to be parsed and executed by the Sybase database on all processing nodes before the table is opened. You can specify a job parameter if required.
  - Close Command: Use it to specify any command, in single quotes, to be parsed and executed by the Sybase database on all processing nodes after the stage finishes processing the table. You can specify a job parameter if required. Ensure column metadata has been specified for the read.
  - Fixed Width: Set this option to True to extract the fields as fixed width.
Performing a lookup on a Sybase database

You can perform two types of lookup on a Sybase database:
- Direct lookup
- In-memory lookup

You select the lookup type by selecting the appropriate from the lookup type from the Sybase enterprise stage properties dialog. The sections below explain these lookup operations.

Performing a direct lookup on a Sybase database table

About this task

To perform a direct lookup on a Sybase database, you must setup the properties of the Sybase enterprise stage appropriately. To perform a direct lookup on a Sybase database, you must first connect to the Sybase enterprise stage and then setup the properties of the stage. Follow the steps below:

- Connect the Sybase enterprise stage to a Lookup stage by using a reference link.
- In the Output Link Properties tab:
  - Set the Lookup Type to Sparse IQ or Sparse ASE.
  - Select the Read DB Type as IQ or ASE. IQ is the default value for this property.
  - Choose a Read Method. This is Table by default (which reads directly from a table), but you can also choose to read by using an auto-generated SQL or user-generated SQL.
  - Specify the table to be read for the lookup.
  - If you are using a Read Method of user-generated SQL, specify the SELECT SQL statement to use. IBM InfoSphere DataStage provides the auto-generated statement as a basis, which you can edit as required.
- Under the Connection category, you can specify a connection details for server, user and password. Alternatively, if you do not want the values to be hard coded, you can insert it from the Job Parameter table or by creating the environment variable either at the job level or at the project level. By default, the InfoSphere DataStage connects to the Sybase default database. However you can specify a different database if required.
- Under the Options category, specify values for the following options:
  - Open Command: Use it to specify a command, in single quotes, to be parsed and executed by the Sybase database on all processing nodes before the table is opened. You can specify a job parameter if required.
  - Close Command: Use it to specify any command, in single quotes, to be parsed and executed by the Sybase database on all processing nodes after the stage finishes processing the table. You can specify a job parameter if required.

Performing an in-memory lookup on a Sybase database table

About this task

This is the default lookup method. To perform an in-memory lookup on a Sybase data, you must follow all the steps in the previous section except you must set the Lookup Type to Normal.
Chapter 8. Sybase IQ12 load stages

This topic describes the following for the Sybase IQ12 load stage:

Functionality of the Sybase IQ12 load stage

The Sybase IQ12 load stage has the following functionality:

- Generation and optional automatic execution of the Sybase IQ, Version 12, commands to load indexes with data from input links.
- New Sybase IQ12 commands follow the SQL standard to use tables instead of indexes. Old Sybase IQ commands are no longer supported.
- Two methods of loading an index set: manual loading or automatic loading by using the Open Client/Open Server (OCOS) or Open Database Connectivity (ODBC).
- Simplified loading of joined index sets that does not require a specific sequence for loading data. You can use the SYNCHRONIZE command to build joined indexes rather than a batch file.
- Support for the server as a stand-alone database without needing a catalog server or direct use of an SQL server. This results in sophisticated indexing and query optimization.
- The ability to load tables during execution of queries.
- Automatic generation of overflow data files if the first data file exhausts physical disk space.
- Support for data files that exceed the 2-GB file size limit for 64-bit file systems.
- The ability to specify Sybase IQ12 commands to be run before and after the insert operation to send diagnostic or verification output to the IBM InfoSphere DataStage log.
- Generation of data files in delimiter-separated ASCII format.
- Support for NLS (National Language Support).

The following functionality is not supported:

- Deletion and recreation of the index set itself.
- Other modes of operation supported in database load utilities, for example, update existing rows only.
- Automatic execution of load commands when the Sybase IQ server resides on a machine other than the one for the InfoSphere DataStage job.
- Loading of joined index sets can only be done by using an after-job subroutine, because the stage instance cannot guarantee that the columns of different tables in the index set are loaded in the correct order. (The job compiler might draw a process boundary through a given stage instance so that there is no one process address space that knows the status of all the links connected to the stage.)
- Use of named pipes to load data.
- Support for loading of Sybase IQ before Version 12.
- Meta data import.
Terminology

The next two sections explain Sybase IQ12 and the Sybase IQ12 Load stage terms used in this document.

Sybase IQ12 terminology

The following table lists the Sybase IQ12 terms used in this document:

Table 18. Sybase IQ12 terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>The Sybase IQ12 command used to delete rows from a table.</td>
</tr>
<tr>
<td>index</td>
<td>A single column in an index set.</td>
</tr>
<tr>
<td>index set</td>
<td>The Sybase IQ12 equivalent of a table. It is a collection of named, typed indexes on columns of data which might have come from a Sybase SQL Server database, a foreign database, or a flat file. The Sybase IQ12 Load stage loads data into index sets. Every index set definition has associated with it a Sybase SQL Server table definition, because Sybase IQ12 uses SQL Server to catalog information about its index sets.</td>
</tr>
<tr>
<td>index space</td>
<td>The Sybase IQ12 entity that contains index sets. The index space owns disk and other resources, and provides a handle for administration.</td>
</tr>
<tr>
<td>IQ Server</td>
<td>The server engine to which you connect in order to use Sybase IQ12. An IQ Server instance provides access to one or more index spaces.</td>
</tr>
<tr>
<td>joined index set</td>
<td>A set of indexes created to allow a relational join between two or more tables.</td>
</tr>
<tr>
<td>LOAD TABLE</td>
<td>The Sybase IQ12 command used to load data into a table.</td>
</tr>
<tr>
<td>SYNCHRONIZE</td>
<td>The Sybase IQ12 command used to synchronize joined indexes.</td>
</tr>
</tbody>
</table>

Sybase IQ12 load terminology

The following table lists the Sybase IQ12 load terms used in this document:

Table 19. Sybase IQ12 terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load stage</td>
<td>A passive stage whose role in an IBM InfoSphere DataStage job is to take streams of tabular data and load them into tables of a target database.</td>
</tr>
<tr>
<td>data file</td>
<td>An ASCII file of row and column data from an input link that is to be loaded. The Sybase IQ12 Load stage generates these files with noncharacter columns separated by vertical bars (</td>
</tr>
<tr>
<td>SQL file</td>
<td>A control file of Sybase IQ12 commands that loads or reloads an index set. An InfoSphere DataStage job generates one SQL file for each input link to each instance of the Sybase IQ12 Load stage. Control files can be executed by piping them to the Sybase dbisql utility.</td>
</tr>
<tr>
<td>stage instance</td>
<td>An individual stage of a given type, appearing as an icon in a job design.</td>
</tr>
</tbody>
</table>
The next two sections describe loading index sets and joined index sets.

Loading index sets

Sybase IQ12 Load supports the following methods using the Load Method property to load the data from its input links into Sybase IQ12 index sets:

- Manual loading
- Automatic loading

Manual loading

The Sybase IQ12 load stage instance generates an ASCII data file and an SQL control file for each input link, but does not load the data into Sybase IQ12. You can load the data later by redirecting the control file to the Sybase dbisql utility. Manual loading is the default.

Automatic loading

The methods to load data automatically are:

- OCOS
- ODBC

Automatic loading works like manual loading. Rows arriving at an input link are written to a data file as in manual loading. When the link reaches the end of the data, appropriate DELETE and LOAD TABLE commands are generated and executed by using a Sybase Client-Library connection to the IQ Server 12. The commands to be executed are also written to an SQL file to log the activity.

OCOS. Automatic loading via OCOS works when both the IBM InfoSphere DataStage server and the IQ Server 12 reside on the same machine. It also works if the following is true:

- The two servers reside on different machines that are connected by a local area network (LAN).
- Open Client is installed on the machine hosting the InfoSphere DataStage server.
- Both machines share a common directory for the output file.

ODBC. Automatic loading via ODBC works if both the InfoSphere DataStage server and IQ Server 12 reside on the same machine. It also works if the following is true:

- The two servers reside on different machines over a LAN.
- The IQ Server 12 client is installed on the machine hosting the InfoSphere DataStage server.
- Both machines share a common directory for the output files.

For more information about overflow directories, see “Disk overflow handling” on page 54.

Loading joined index sets

Joined index set loading has been simplified. Since the data for each joined index set no longer needs to be loaded in a specific sequence, you can load joined indexes automatically. You need to explicitly execute a SYNCHRONIZE command after job completion. You can use the Post-insert Command property or ExecDOS to do this.
The data for each table in the join must be synchronized to allow you to bring the
joined indexes up to date making them available for queries to use.

Support for joined indexes in the Sybase IQ12 Load stage is complicated because
the input links to a stage instance are not guaranteed to run in the same process.
Depending on the overall design of the job, the IBM InfoSphere DataStage job
compiler might draw process boundaries through an Sybase IQ12 Load stage
instance. This makes it impossible to know while the job is running when the last
link has closed. Consequently, the actual synchronizing of data in a joined indexes
has to be done outside the job itself. You can execute the SYNCHRONIZE
command as an ExecDOS after-job subroutine to load and synchronize joined
indexes.

**Loading and synchronizing joined index sets**

**About this task**

To load and synchronize joined index sets:

**Procedure**

1. Open the job in the InfoSphere DataStage Designer Client GUI.
2. Choose Edit > Job Properties to display the Job Properties dialog box. From
   here you can configure job parameters.
3. On the General page, select ExecDOS from the After-job subroutine list.
4. Enter the following command in the Input Value field to run SYNCHRONIZE
   against the appropriate tables:
   
   ```
   dbisql -q -c Userid=dba; Password=sql;ServerName=asiqdemo;DatabaseName=asiqdemo;
   SYNCHRONIZE JOIN INDEX emp_dept_join1, emp_dept_join2
   ```

**Results**

Although the synchronizing must be done outside the job itself, the actual loading
of a joined index can be done by using any of the loading method. Since the data
for each joined index no longer needs to be loaded in a specific sequence, it is
possible to load joined indexes by using the automatic loading. You must explicitly
execute a SYNCHRONIZE command after the job completes.

**Disk overflow handling**

The Sybase IQ12 Load stage must be able to handle load operations in the
multiple-gigabyte range. These large data sets can exhaust the free space on the
disk drive or partition that is receiving the data file.

You can use a semicolon-separated list of directory paths as the value of the
Output Path property to handle these situations. If the stage runs out of disk space
during a job run and cannot write a row to the data file, it opens an overflow data
file in the second directory in the list and continues. In this way, the data can be
spread among multiple disk drives or partitions.

Add the pathnames of overflow files to the FROM clause of the LOAD TABLE
command that loads the data.
Stage and link properties

The Sybase IQ12 load stage supports stage and link properties that are visible from the InfoSphere DataStage Designer. You need to supply values for these properties in the stage grid-style editor.

The tables in the next two sections include the following column heads:

- **Prompt** is the text that you see in the stage editor user interface.
- **Type** is the data type of the property.
- **Default** is the value used if you do not supply a value.
- **Description** gives details about the properties.

### Stage properties

Use this tab to specify connection details.

The Sybase IQ12 Load stage supports the following stage properties. The stage properties are listed in the logical order in which you design a job.
<table>
<thead>
<tr>
<th>Prompt</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Method</td>
<td>List</td>
<td>Manual</td>
<td>The method in which the data is to be loaded into the Sybase IQ12 tables. Valid methods are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Manual.</strong> Generates the SQL and data files which must be manually loaded using <code>dbisql</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Autoload via OCOS.</strong> Automatically loads the data using Open Client/Open Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Autoload via ODBC.</strong> Automatically loads the data using ODBC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Automatic loading tells the Sybase IQ12 Load stage to connect to the IQ server and execute the commands in the SQL file after the last row of data has been written to the corresponding data file. If this option is used for joined indexes, you must explicitly execute <code>SYNCHRONIZE</code> commands to update the joined index. (Manual/Autoload via OCOS/Autoload via ODBC)</td>
</tr>
</tbody>
</table>
### Table 20. Stage Properties (continued)

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQ Server/Datasource Name</td>
<td>String</td>
<td>None</td>
<td>The name of the IQ12 Server or ODBC data source as defined for any IQ client program. On Windows platforms, Autoload using OCOS uses the name of the IQ12 Server defined in <code>sql.ini</code>. On UNIX platforms, Autoload using OCOS uses the name of the IQ12 Server defined in <code>$sybase/interfaces</code>. Autoload using ODBC uses the name of the ODBC database. Required for autoloads.</td>
</tr>
<tr>
<td>IQ Database Name</td>
<td>String</td>
<td>None</td>
<td>The name of the target Sybase IQ12 database. This name appears as the argument to a connection request. Required for autoloads.</td>
</tr>
<tr>
<td>IQ User ID</td>
<td>String</td>
<td>None</td>
<td>The Sybase IQ12 user name used when connecting to the IQ12 Server to perform the load. Required for autoloads.</td>
</tr>
<tr>
<td>IQ Password</td>
<td>String</td>
<td>None</td>
<td>The Sybase IQ12 password used when connecting to the IQ12 Server to perform the load. Required for autoloads.</td>
</tr>
<tr>
<td>Output Path</td>
<td>String</td>
<td>C:	emp</td>
<td>A directory where the Sybase IQ12 Load stage creates SQL and data files. SQL files are lists of IQ12 commands. Data files are ASCII files of data from input links.</td>
</tr>
</tbody>
</table>
## Link properties

Use the **Properties** tab to specify the load operation.

The Sybase IQ12 Load stage supports the following input link properties, which you use to write to Sybase tables. The properties are listed in the logical order in which you design a job.

### Table 21. Link Properties

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name</td>
<td>String</td>
<td>None</td>
<td>The name of the target Sybase IQ12 table to be loaded with data from this link. This name is the argument of the LOAD TABLE command. Required.</td>
</tr>
<tr>
<td>SQL File Name</td>
<td>String</td>
<td>Tablename.sql</td>
<td>The SQL file (generated by the Sybase IQ12 Load stage) containing deletion and insertion commands to run at database load time. This file is always created for documentation purposes, even if the IQ12 commands are executed using OCOS or ODBC. Its name defaults to the table name, with an extension of .sql. The SQL file is always created in the first directory in &quot;Output Path.&quot;</td>
</tr>
</tbody>
</table>
### Table 21. Link Properties (continued)

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data File Name</td>
<td>String</td>
<td><em>Tablename.dat</em></td>
<td>The file name of a flat ASCII output file generated by Sybase IQ12 Load containing the rows and columns of data to load into the table for the link. The file name defaults to the table name, with an extension of <code>.dat</code>. The file capacity is as large as the disk space permits. Choose disk space that accommodates a single output file. If you are running a job created in a previous version of Sybase IQ12 Load, be sure that the file accommodates the size of the data.</td>
</tr>
<tr>
<td>CHAR Delimiter</td>
<td>String</td>
<td>*</td>
<td>(vertical bar)*</td>
</tr>
<tr>
<td>Clear Before Load</td>
<td>List</td>
<td>Yes</td>
<td>Controls whether a DELETE command is generated before the LOAD command that loads the new data into the table. For single tables, an DELETE command is generated. In both cases, the FROM and WHERE clauses in the corresponding properties are added to the command. (Yes/No)</td>
</tr>
</tbody>
</table>
Table 21. Link Properties (continued)

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE...FROM</td>
<td>String</td>
<td>None</td>
<td>This permits full specification of the FROM table-list clause of the DELETE command. Defaults to the table name (the value of the TABLE property for the link). The keyword FROM is not required. The absence of a FROM clause is equivalent to the TRUNCATE command.</td>
</tr>
<tr>
<td>DELETE... WHERE</td>
<td>String</td>
<td>None</td>
<td>Specifies optional search_conditions for the DELETE command. If defined, the search_conditions go into the WHERE clause of the command. The keyword WHERE is not required.</td>
</tr>
<tr>
<td>LOAD TABLE... FORMAT</td>
<td>List</td>
<td>ascii</td>
<td>Specifies the optional format option for the LOAD TABLE command. (ascii/binary)</td>
</tr>
<tr>
<td>LOAD TABLE... STRIP</td>
<td>List</td>
<td>ON</td>
<td>Specifies the optional strip option for the LOAD TABLE command. (ON/OFF)</td>
</tr>
<tr>
<td>LOAD TABLE... CHECKPOINT</td>
<td>List</td>
<td>OFF</td>
<td>Specifies the optional checkpoint option for the LOAD TABLE command. (ON/OFF)</td>
</tr>
<tr>
<td>LOAD TABLE... LOADOPTIONS</td>
<td>String</td>
<td>None</td>
<td>Specifies load options for the LOAD TABLE command.</td>
</tr>
<tr>
<td>Prompt</td>
<td>Type</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Pre-insert Command</td>
<td>String</td>
<td>None</td>
<td>The full literal text of an optional IQ12 command to run before the DELETE and LOAD TABLE commands that make up the load. This can be used to run IQ12 diagnostic commands. Output from these commands appears in the DataStage job log during automatic loading.</td>
</tr>
<tr>
<td>Post-insert Command</td>
<td>String</td>
<td>None</td>
<td>The full literal text of an optional IQ12 command to run after the DELETE and LOAD TABLE commands that make up the load. This can be used to run IQ12 diagnostic commands. Output from these commands appears in the DataStage job log during automatic loading.</td>
</tr>
</tbody>
</table>
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:

**wsetsrc**[-S server] [-l label] [-n name] source

The source argument is the only required argument for the **wsetsrc** 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.

**wrb** -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 22. 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>
<tr>
<td>IBM representatives</td>
<td>You can contact an IBM representative to learn about solutions at <a href="http://www.ibm.com/connect/ibm/us/en/">www.ibm.com/connect/ibm/us/en/</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/inforcenter/topic/com.ibm.swg.im.iis.productization.iisinfohome.doc/ic-hompage.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/inforcenter/topic/com.ibm.swg.im.iis.productization.iisinfohome.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/inforcenter/iisinfsrv/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/.
- 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/
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\(^\text{\textregistered}\), ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACS\(^\text{\textregistered}\) licensee of the United States Postal Service.

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

A
after-load stored procedures 28

B
BCP (Bulk Copy Program) utility
bulk copy API 6
description 6, 25
running from command line 6
switches 6, 25
BCPLoad stages
Columns tab 31
configuration requirements 26
ingeneral tab 26
Inputs page 31
Stage page 27
Inputs page 31
introduction 6
metadata, importing 26
NLS tab 30
overview 25
Properties tab 27
SQL data types 26
Stage page 26
stored procedures 28
table definitions 26
before-load stored procedures 28
bulk copy program, see BCP utility 6, 25

C
code set maps, defining
BCPLoad stages 30
Sybase OC stages 13
client libraries
CTLIB 7
DBLIB 7
NetLIB 7
command-line syntax conventions 65
commands
syntax 65
customer support contacting 69

D
data Browser 15, 18
dynamic access 5

I
index sets 53, 54

L
legal notices 73
libraries, client 7

N
NetLIB client library 7

P
product accessibility
accessibility 63
product documentation
accessing 71

R
remote Sybase IQ server
loading data 34

S
setting environment variables for databases
setting 2, 3
software services
contacting 69
special characters in command-line syntax 65
SQL data types
BCPLoad stages 26
Sybase OC stages 20
SQL Server
unsupported BCP switches 25
SQL statements
syntax 19
stored procedures
BCPLoad stages 28
Stage page 35
storage procedures 28
Sybase BCP Load stages, see BCPLoad stages 25
Sybase connectivity, description 5
Sybase Enterprise stages
Advanced tab
Input page 42
Output page 44
Stage page 35
Columns tab
Input page 42
Output page 44
configuration requirements 8
direct lookup 49
Editor 35
gen general tab
Input page 36
Output page 42
Stage page 35

Sybase Enterprise stages (continued)
loading data 34

Sybase IQ12 Load stages
configuration requirements 6
disk overflow 54
functionality 51
index sets 53, 54
introduction 6
link properties 58
loading index sets 53
automatically 53
manually 53
OCOS method 53
ODBC method 53
OCOS method 53
Sybase IQ12 52
Sybase IQ12 Load 52
Sybase OC stages
AfterSQL statements 17
BeforeSQL statements 16
character set mapping 13
configuration requirements 6
connection parameters 12
data type support 20, 23
approximate data 22
binary data 23
categorical data 20
dates 23
decimal data 21
integer data 21
money 22
FROM clause 19
General tab
Input page 13
Output page 17
generated queries 19
generated SQL statements 15
GROUP BY clause 19
HAVING clause 19
input data, defining 13
Input page 13, 15
introduction 5
output data, defining 17
Output page 11, 17

© Copyright IBM Corp. 1998, 2013
Sybase OC stages (continued)
  packet size  12
  SELECT clause  19
  Stage page  11
  stored procedures  24
  SYBASEOC stage dialog box  11
  transaction isolation level  12
  user-defined queries  20
  user-defined SQL statements  16
  WHERE clause  19
Sybase Server
  unsupported BCP switches  25
syntax
  command-line  65

T
  trademarks
    list of  73

W
  web sites
    non-IBM  67