

# **Technical Bulletin**

**Part No. 74-0136**

## **DataStage Informix CLI**

This technical bulletin describes Release 1.3 of the DataStage Informix CLI stage, formerly known as the DataStage Informix Call Interface Plug-in. This stage reads and writes data to and from any DataStage stage into any Informix database. It also provides native data browsing and meta data import from the Informix database to DataStage.

© 1999–2003 Ascential Software Corporation. All rights reserved. Ascential, Ascential Software, DataStage, MetaStage, MetaBroker, and Axielle are trademarks of Ascential Software Corporation or its affiliates and may be registered in the United States or other jurisdictions. Adobe Acrobat is a trademark of Adobe Systems, Inc. HP and Tru64 is either a registered trademark or trademark of Hewlett-Packard Company. AIX, DB2, DB2 Universal Database, IBM, Informix, MQSeries, Red Brick, UniData, UniVerse, and WebSphere are either registered trademarks or trademarks of IBM Corporation. Microsoft, Windows, Windows NT, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Teradata is a registered trademark of NCR International, Inc. Oracle, Oracle8i, and Oracle 9i are either registered trademarks or trademarks of Oracle Corporation. Solaris, Sun, and Sun Microsystems are either trademarks or service marks of Sun Microsystems, Inc. Adaptive Server, Open Client, and Sybase are either registered trademarks or trademarks of Sybase, Inc. Linux is a trademark of Linus Torvalds. WinZip is a registered trademark of WinZip Company, Inc. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. Other marks mentioned are the property of the owners of those marks.

This product may contain or utilize third party components subject to the user documentation previously provided by Ascential Software Corporation or contained herein.

## Printing History

First Edition (74-0136) for Release 1.0, June 1999  
Second Edition (74-0136) for Release 1.1, August 1999  
Third Edition (74-0136) for Release 1.2, May 2000  
Third Edition (74-0136) updated for Release 1.2, December 2000  
Updated (74-0136) for Release 1.3, December 2001  
Updated (74-0136) for Release 1.3, March 2002  
Updated (74-0136) for Release 1.3, August 2002  
Updated (74-0136) for Release 1.3, August 2003

## How to Order Technical Documents

To order copies of documents, contact your local Ascential subsidiary or distributor, or call our main office at (508) 366-3888.

Documentation Team: Marie E. Hedin

## Introduction

This technical bulletin describes the following for Release 1.3 of Informix CLI updated for DataStage Release 7.0:

- Functionality
- Configuration requirements
- Installation
- Terminology
- Defining the Informix CLI connection
- Defining Informix CLI input data
- Writing data to Informix
- Defining Informix CLI output data
- Reading data from Informix
- Mapping data types
- Stored procedure support
- Properties

Currently, you can only use the ODBC stage to access Informix data. The Informix CLI stage is the Informix implementation of the Microsoft Open Database Connectivity (ODBC) standard. It enables DataStage to read and write data to and from Informix databases using the Informix CLI native programming interface. Informix CLI offers the following advantages over the DataStage ODBC stage:

- Increased processing speed
- Simplified configuration on UNIX platforms
- Support for Versions 7, 8, and 9 of the Informix server software
- Advanced support for target table DDL (Create and Drop Table)
- Native meta data import

The Informix CLI stage lets you connect and process SQL statements in the native Informix environment. Because the CLI programming API is a native Informix implementation of the ODBC standard, Windows users (see [note](#) below) still must use an ODBC driver manager to access CLI and use the ODBC Data Source Administrator to configure data sources.

In summary, the Informix CLI stage lets you do the following for a target Informix database:

- Read and write data
- Create and drop tables
- Import table and column definitions
- Browse native data with the custom GUI

The Informix CLI stage lets DataStage read and write data to and from an Informix database. Each Informix CLI 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 an Informix database. You can specify the data on an input link using an SQL statement generated by DataStage or a user-constructed statement.
- **Output links.** Specify the data you are extracting, which is a stream of rows to be read from an Informix database. You can specify the data on an output link using an SQL SELECT statement generated by DataStage or a user-constructed statement.
- **Reference output links.** Each link represents rows that are key read from an Informix database (using the key columns in a WHERE clause of the SELECT statement that is constructed by DataStage or user-specified).

**Note:** The term Windows is used to represent the following:

- Windows NT
- Windows 2000
- Windows Server 2003

## Functionality

Informix CLI has the following functionality:

- Stream input, stream output, and reference output links.
- The ability to use the **Derivation** field to specify fully qualified column names for output links. (If the ODBC stage precedes the Informix CLI stage in a job flow, you can also use the **Derivation** field to specify fully qualified column names for input links.)
- The ability to import table and column definitions from the target Informix database and store them in the DataStage Repository. For more information about meta data import, see DataStage documentation.
- NLS (National Language Support). For more information, see *DataStage NLS Guide*.
- Reject row handling.
- File names to contain your SQL statements.

- Support of MetaStage. For more information, see *MetaStage User's Guide*.
- Stored procedures.
- Data browsing, which is the ability to use the custom GUI for the stage to view sample native table data residing on the target Informix database.

The following functionality is not supported:

- Bulk loading of Informix tables from stream input. Continue to use either the Informix or Informix XPS loaders for bulk loading into an Informix database.
- Replacing the ODBC stage. The Informix CLI stage does not replace the ODBC stage. DataStage users who created jobs using the ODBC stage to access an Informix database may continue to run these jobs.
- Non-ANSI SQL constructs in stage-generated SQL statements.
- Version-specific SQL constructs in stage-generated SQL statements.
- Text and byte data types.

## Configuration Requirements

For general configuration requirements, see *DataStage Plug-In Installation and Configuration Guide*.

### UNIX Notes

If the data source uses a translation DLL, you must add INFORMIXDIR/lib/esql to the shared library search path in *dsenv*. If you do not, and your data source requires a translation DLL, the message "Unable to load translation DLL" appears in the DataStage job log.

Informix CLI looks for an *.odbc.ini* file in the *dshome* directory, the DataStage server engine home directory that is stored in */.dshome*. (To see the location of the DataStage installation directory, enter `$cat 'cat /.dshome'`.)

The driver field in the data source entry for the *.odbc.ini* file *must* reference the DataDirect 4.0 non-wired ODBC driver that is included on the DataStage CD. Other drivers are *not* supported.

Tru64 requires the Informix CLI ODBC driver.

**Informix Connection Examples.** The following example shows sample DSN entries for AIX and HP platforms for DataStage 7.0:

```
[Informix]
Driver=/u4/dsadm/Ascential/DataStage/branded_odbc/lib/VMinf19.so
Description=DataDirect Informix
Database=<stores_demo>
LogonID=<userid>
Password=<password>
ServerName=<informixserver>
HostName=<informixhost>
Service=<online>
Protocol=ontlitcp
EnableInsertCursors=0
GetDBListFromInformix=0
CursorBehavior=0
CancelDetectInterval=0
TrimBlankFromIndexName=1
ApplicationUsingThreads=1
```

The following example shows sample DSN entries for Solaris platforms for DataStage 7.0:

```
[Informix]
Driver=/u4/dsadm/Ascential/DataStage/branded_odbc/lib/VMinf19.so
Description=DataDirect Informix
Database=<stores_demo>
LogonID=<userid>
Password=<password>
ServerName=<informixserver>
HostName=<informixhost>
Service=<online>
;Protocol=ontlitcp
EnableInsertCursors=0
GetDBListFromInformix=0
CursorBehavior=0
CancelDetectInterval=0
TrimBlankFromIndexName=1
ApplicationUsingThreads=1
```

The following example shows sample DSN entries for Tru64 and Linux platforms for DataStage 7.0:

```
[hpds_stores]
Driver=/u1/informix/lib/cli/iclis09b.so
Description=INFORMIX 3.3 32-BIT
Database=<stores7>
LogonID=<userid>
pwd=<password>
Servername=<hpds.1>
CursorBehavior=0
CLIENT_LOCALE=en_us.8859-1
DB_LOCALE=en_us.8859-1
TRANSLATIONDLL=/u1/informix/lib/esql/igo4a304.so
```

Every Informix data source to which your DataStage jobs connect must have an entry in the *.odbc.ini* file. The only required fields in the data source specification are the **Database** and **Server name**. If you choose to include the login ID (UID) and/or password (PWD), you can leave the **User Name** and **Password** properties blank. If you enter values for these properties, the values in *.odbc.ini* are ignored. For more information on the format of the *.odbc.ini* file, see *Informix CLI Programmer's Manual*.

You can also use this *.odbc.ini* file for other ODBC applications including DataStage jobs using the ODBC stage.

## Installing the Plug-In

For instructions and information supporting the installation, see *DataStage Plug-In Installation and Configuration Guide*.

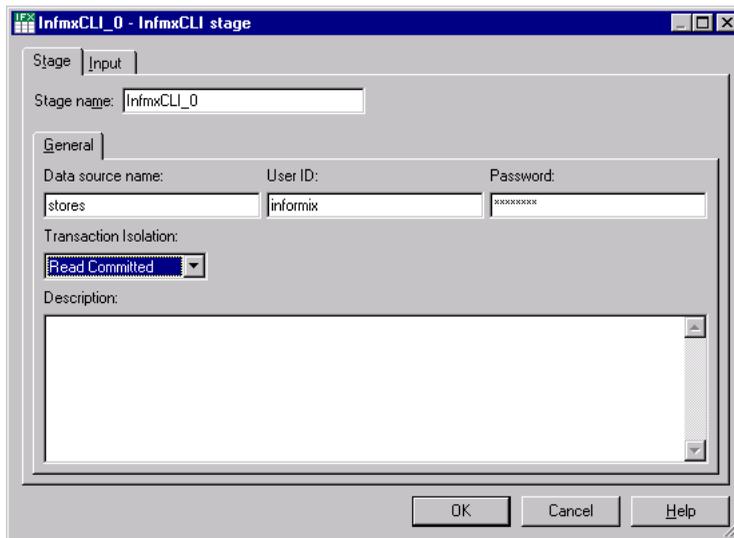
## Terminology

The following table lists the Informix CLI terms used in this document:

<b>Term</b>	<b>Description</b>
AD/XP	Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options 8.n.
CLI	Informix Call Interface. This is the Informix implementation of the Microsoft Open Database Connectivity (ODBC) standard.
IDS	Informix Dynamic Server Version 7.n.
W/D	Informix Dynamic Server, Workgroup and Developer Edition.

## Defining the Informix CLI Connection

When you use the GUI to edit an Informix CLI stage, the **InfmxCLI Stage** dialog box appears:



This dialog box has the **Stage**, **Input**, and **Output** 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 Informix data source, login information, and transaction isolation level information for concurrency control in jobs. You can enter text to describe the purpose of the stage in the **Description** field. The properties on this tab define the connection to the Informix data source. For details, see [“Connecting to an Informix Data Source”](#) on page 7.

The **NLS** tab defines a character set map to be used with the stage. This tab appears only if you have installed NLS for DataStage. For details, see [“Defining Character Set Mapping”](#) on page 9.

**Note:** You cannot change the name of the stage from this dialog box. For details on changing stage names, see DataStage documentation.

- **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. It also specifies how data is written and contains the SQL statement or call syntax used to write data to an Informix table. It also

specifies how to create the target table if desired and how to drop it if necessary.

- **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 contains the SQL SELECT statement or call syntax used to read data from one or more Informix tables or views.

The main phases in defining an Informix CLI stage from the **InfmtxCLI Stage** dialog box are as follows:

1. Connect to an Informix data source (see [page 7](#)).
2. Optional. Define a character set map (see [page 9](#)).
3. Define the data on the input links (see [page 9](#)).
4. Define the data on the output links (see [page 18](#)).

## Connecting to an Informix Data Source

The Informix connection parameters are set on the **General** tab on the **Stage** page. To connect to an Informix data source:

1. Enter the name of the Informix data source to access in the **Data source name** field. There is no default. You can enter any valid data source although unpredictable results may occur if the data source is not an Informix data source. This field is required.

**Windows.** Define data sources using the ODBC Administrator.

**HP-UX 11.0 UNIX platform.** Define data sources (DSN) for Informix databases in the *.odbc.ini* file.

For more information about defining data sources, see *Informix CLI Programmer's Manual*.

2. Enter the name to use to connect to the Informix data source in the **User ID** field. The Informix CLI connection also uses operating system authentication or the *.odbc.ini* data source definition on UNIX platforms for connection to the database. For more information about the *.odbc.ini* file, see the UNIX Platforms section in "[Configuration Requirements](#)" on page 3.

**Note:** You cannot use an ODBC stage that uses an Informix database as a source or target in the same job with another stage that uses Informix CLI.

Without a user name in the **User ID** field, the Plug-in tries to use a user name provided in the data source definition. Without this user name, the operating system (OS) authentication uses the OS user name of the person running the DataStage job. However, if the Plug-in is running on an HP platform, it does not default to the OS user name of the person running the DataStage job.

This user must have sufficient privileges to access the specified database and source and target tables.

3. Enter the password that is associated with the specified user name to use in the **Password** field. For security, it displays asterisks instead of the value you enter. There is no default.

For platforms other than HP, if no user name is provided in the **User ID** field, leave **Password** blank in order to use default authentication (data source definition or OS authentication).

4. Choose an appropriate transaction isolation level to use from the **Transaction Isolation** drop-down list box. This level provides the necessary concurrency control between transactions in the job and other transactions.

Use one of the following transaction isolation levels:

- **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.
- **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. This is the default.
- **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.

**Note:** Transaction isolation levels apply only to databases that support logging. Otherwise, they are ignored.

5. Optional. Describe the purpose of the Informix CLI stage in the **Description** field.

## Defining Character Set Mapping

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

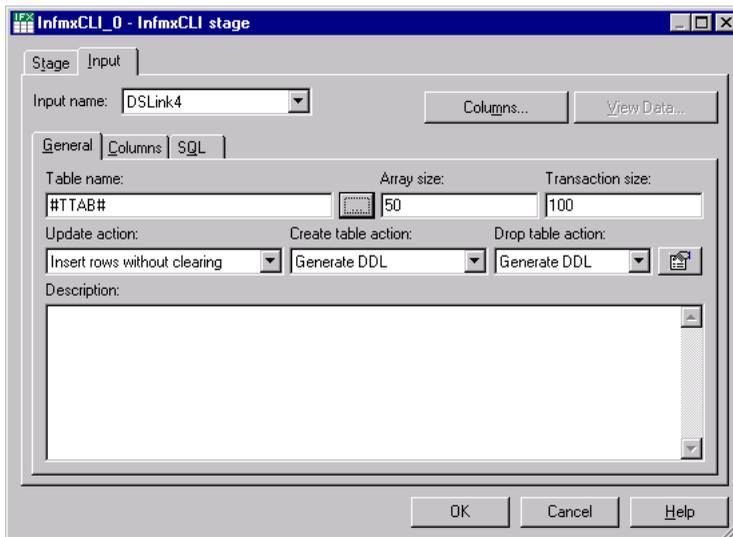
Specify information using the following button and fields:

- **Map name to use with stage.** 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 list.
- **Use Job Parameter...** Specifies 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.
- **Show all maps.** Lists all the maps that are shipped with DataStage.
- **Loaded maps only.** Lists only the maps that are currently loaded.

For more information about NLS or job parameters, see DataStage documentation.

## Defining Informix CLI Input Data

When you write data to a table in an Informix database, the Informix CLI stage has an input link. Define the properties of this link and the column definitions of the data on the **Input** page in the **InfmxCLI Stage** dialog box of the GUI.



The **Input** page has an **Input name** field, the **General**, **Columns**, and **SQL** tabs, and the **Table Properties** (at the right of the **Drop table action** list box), **Columns...**, and **View Data...** buttons:

- **Input name.** The name of the input link. Choose the link you want to edit from the **Input name** drop-down list box. This list box displays all the input links to the Informix CLI stage.
- Click the **Columns...** button to display a brief list of the columns designated on the input link. As you enter detailed meta data in the **Columns** tab, you can leave this list displayed.
- Click the **View Data...** button to open the Data Browser. This lets you look at the data associated with the input link. For a description of the Data Browser, see DataStage documentation.

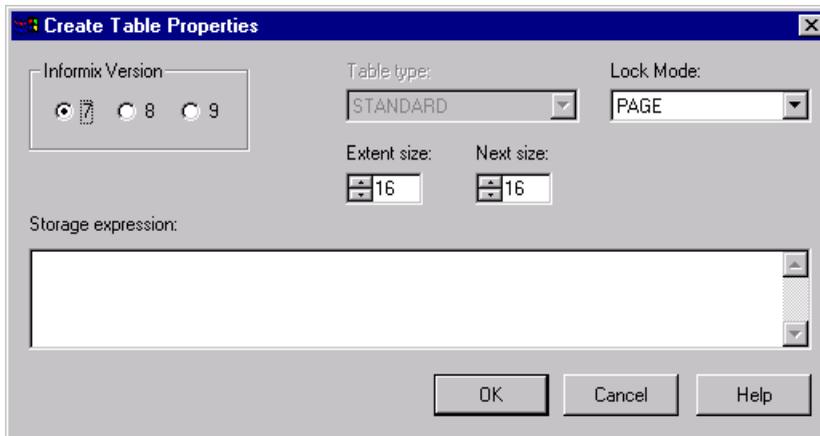
**General Tab.** This tab is displayed by default.

It contains the following fields:

- **Table name.** The name of the target table to update. You must specify **Table name** if you do not specify **User-defined SQL**. There is no default. This field is editable when the update action is *not* **User-defined SQL** (otherwise, it is read-only). You can also click the ... button at the right of the **Table name** field to browse the Repository to select the table.
- **Array size.** The number of rows written to the database at one time. The default is 50 rows that are cached before being written to the database.
- **Transaction size.** The number of rows that the stage processes before committing a transaction to the database. The transaction size should always be a multiple of the array size. The default is 100. This field is ignored for nonlogging databases.
- **Update action.** Specifies which stage-generated SQL statements are used to update the target table. Some update actions require key columns to update or delete rows. The default is to insert rows without clearing. Choose one of the following options:
  - **Insert rows without clearing.** Inserts the new rows in the table.
  - **Clear the table, then insert rows.** Deletes the contents of the table before inserting the new rows, with slower performance because of transaction logging.
  - **Delete existing rows only.** Deletes existing rows in the target file that have identical keys in the input rows.

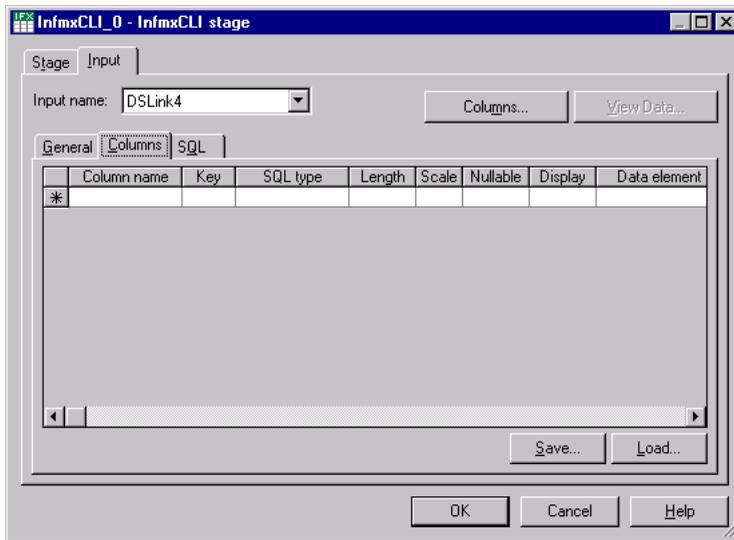
- **Replace existing rows completely.** Deletes the existing rows in the target table that have the same key as the new rows, then adds the new rows to the table.
- **Update existing rows only.** Updates the existing data rows. Any rows in the data that do not exist in the table are ignored.
- **Update existing or insert new rows.** Updates the existing data rows before inserting 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 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 using a user-defined SQL statement. When you select this option, it overrides the default SQL statement generated by the stage. If you choose this option, you enter data on the **SQL** tab. See [“Using User-Defined SQL Statements”](#) on page 16 for details on how to do this.
- **Create table action.** Choose one of the following options to create the target table in the specified database:
  - **Do not create target table.** Specifies that the target table is not created, and the **Drop table action** field and the **Table Properties** button (at the right of the field) are disabled.
  - **Generate DDL.** Specifies that the stage generates the CREATE TABLE statement using information obtained from “Target Table,” the column definitions grid, and the advanced table properties (see the **Table Properties** button later in this section).
 

**User-defined DDL.** Specifies that you enter the appropriate CREATE TABLE statement on the **SQL** tab described on [page 14](#).
- **Drop table action.** Lets you control the dropping of the target table before it is created by the stage. If you choose not to create the target table, this field is disabled. The list box displays the same items as the **Create table action** list box except that they apply to the DROP TABLE statement.
- **Table Properties button.** Click the button at the right of the **Drop table action** list box to display the **Create Table Properties** dialog box. You can then specify the following advanced table properties from this dialog box.



- **Informix Version.** Specifies the version of the target Informix database. This property lets the interface display interface elements that are relevant to the database version. For example, **Table type** applies only for Version 8, so it is disabled for Versions 7 or 9.
- **Table type.** Choose one of the following uneditable options: RAW, OPERATIONAL, STANDARD, or STATIC. This field is available only for Informix Version 8.
- **Lock Mode.** Choose one of the following uneditable options: PAGE, TABLE, or ROW. Row locking is available only for Informix Version 8.
- **Extent size and Next size.** These buttons accept an integer. The GUI restrains the lower limit of these values.
- **Storage expression.** Lets you enter a storage expression that is appropriate for the target database. It corresponds to the Storage Options clause described in *Informix Guide to SQL Syntax*. No validation is performed.
- **Description.** Optionally, enter text to describe the purpose of the link.

**Columns Tab.** This tab contains the column definitions for the data written to the table or file.



The **Columns** tab behaves the same way as the **Columns** tab in the ODBC stage. For a description of how to enter and edit column definitions, see DataStage documentation.

The column definitions are used in the same way and order they appear in the Columns meta data definitions grid except for the **Description** field. The Informix CLI stage uses the text in this field to define the column precision for Informix DATETIME columns when you want the target table to be created. The **Description** field must be empty for a DATETIME column with default precision.

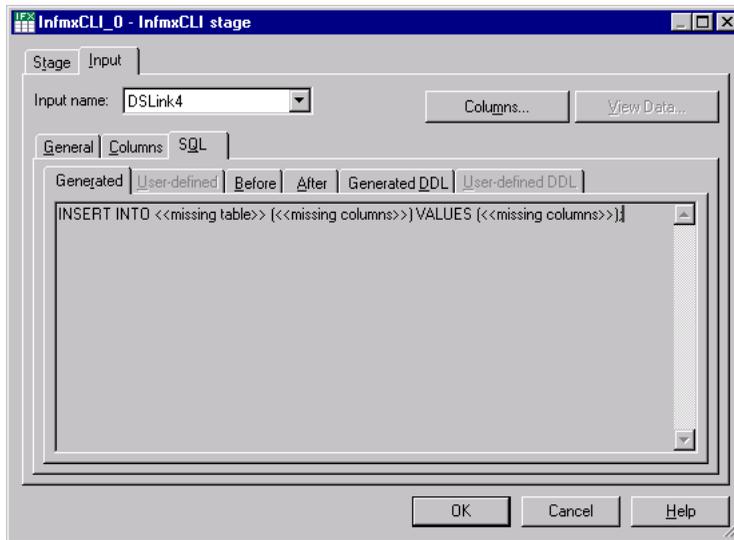
Columns defined as DataStage Time are always created in Informix as "DATETIME hour to second." DataStage Timestamp columns are created in Informix as DATETIME. Use the **Description** field to specify the Informix DATETIME precision if you do not want to use the default for the database.

For example, suppose call\_dtime is a column name for the SQL Timestamp data type. Since Informix uses the DATETIME data type to represent dates and times, a conversion must be made between the DataStage representation and the Informix representation. Informix allows DATETIME columns to be defined with a precision that represents a full or contiguous subset of the components of a timestamp. This DATETIME precision specification is given when the table is created.

When the Informix CLI stage generates the CREATE TABLE DDL statement, it uses the value specified in the **Description** column to define the DATETIME precision. Specify **year to day** in the **Description** column to cause call\_dtime to be

created as a DATETIME column that accepts dates in the 1997-04-22 format. Specify **year to fraction(3)** to cause res\_dtime to be created as a DATETIME column accepting full ANSI timestamps.

**SQL Tab.** This tab contains the **Generated**, **User-defined**, **Before**, **After**, **Generated DDL**, and **User-defined DDL** tabs.



Use these tabs to display the stage-generated SQL statement and the SQL statement that you can enter.

- **Generated.** Displays the SQL statements constructed by DataStage that are used to write data to Informix. The statements represent the uneditable result of the selection made in the **Update action** field on the **General** tab. You can use **Copy** to copy them to the Clipboard for use elsewhere. This tab is displayed by default.
- **User-defined.** Select **User-defined SQL** from the **Update action** field on the **General** tab to enable this tab. The GUI displays the stage-generated SQL statement on this tab as a starting point. However, you can enter any valid, appropriate SQL statement. The box size changes proportionately when you resize the main window to display long SQL statements.
- **Before.** This tab contains the SQL statements executed before the stage processes any job data rows. The elements on this tab correspond to the "Before SQL" and "Continue if Before SQL failed" grid properties. The **Before** and **After** tabs look alike. The "Continue if Before SQL fails" and

“Continue if After SQL fails” properties are represented by check boxes and the SQL statement is entered in a resizable edit box.

- **After.** This tab contains the SQL statements executed after the stage processes job data rows. The elements on this tab correspond to the “After SQL” and “Continue if After SQL failed” grid properties. The **Before** and **After** tabs look alike. The “Continue if Before SQL fails” and “Continue if After SQL fails” properties are represented by check boxes and the SQL statement is entered in a resizable edit box.
  - **Generated DDL.** Select **Generate DDL** or **User-defined DDL** from the **Create table action** field on the **General** tab to enable this tab. The **CREATE statement** field displays the CREATE TABLE statement that is generated from the column meta data definitions and the information provided on the **Create Table Properties** dialog box. If you select an option other than **Do not drop target table** from the **Drop table action** list, the **DROP statement** field displays the generated DROP TABLE statement for dropping the target table.
  - **User-defined DDL.** Select **User-defined DDL** from the **Create table action** or **Drop table action** field on the **General** tab to enable this tab. The generated DDL statement is displayed as a starting point to define a CREATE TABLE and a DROP TABLE statement.

The **DROP statement** field is disabled if **User-defined DDL** is not selected from the **Drop table action** field. If **Do not drop target** is selected, the **DROP statement** field is empty in the **Generated DDL** and **User-defined DDL** tabs.

**Note:** Once you modify the user-defined DDL statement from the original generated DDL statement, changes made to other table-related properties do not affect the user-defined DDL statement. If, for example, you add a new column in the column grid after modifying the user-defined DDL statement, the new column appears in the generated DDL statement but does not appear in the user-defined DDL statement.

## Writing Data to Informix

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

## Using Generated SQL Statements

By default, DataStage writes data to an Informix table using an SQL INSERT, DELETE, or UPDATE statement that it constructs. The generated SQL statement is automatically constructed using the DataStage 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:

1. Enter a table name in **Table name** on the **General** tab on the **Input** page.
2. Specify how you want the data to be written by choosing an option from the **Update action** drop-down list box.
3. [“Defining Informix CLI Input Data”](#) on page 9 for a description of the update actions.
4. Enter an optional description of the input link in the **Description** field.
5. Click the **Columns** tab on the **Input** page.
6. Edit the Columns grid to specify column definitions for the columns you want to write.

The SQL statement is automatically constructed using your chosen update action and the columns you have specified. You can now optionally view this SQL statement.

7. 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.
8. Click **OK** to close this dialog box. Changes are saved when you save your job design.

## Using User-Defined SQL Statements

Instead of writing data using an SQL statement constructed by DataStage, you can enter your own SQL INSERT, DELETE, or UPDATE statement for each Informix CLI 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:

1. Choose **User-defined SQL** from the **Update action** drop-down list box from the **General** tab of the **Input** page.
2. Click **User-defined** tab on the **SQL** page. By default you see the stage-generated SQL statement. You can edit this statement or enter the SQL statement you want to use to write data to the target Informix 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 parameter markers ( ? ) 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, each is executed as a separate transaction. Terminate individual SQL statements with a semicolon (;).

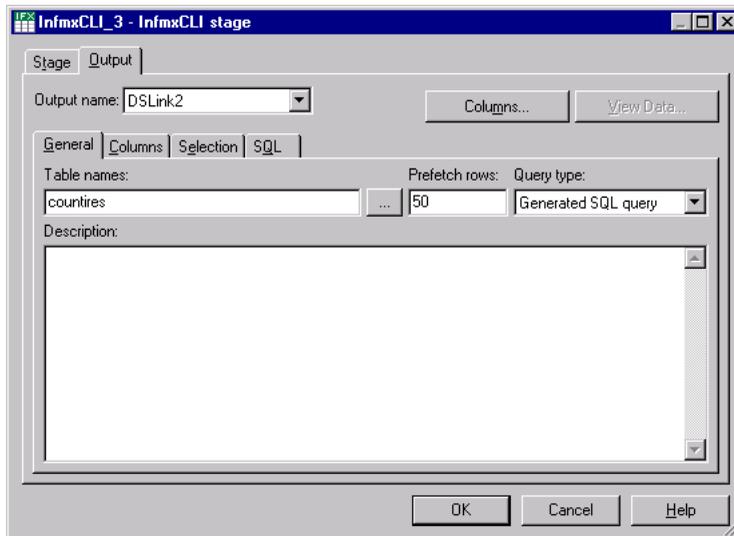
Like the **Storage expression** edit box on the **Create Table Properties** dialog box, the size of this box changes proportionately when the main window is resized in order to allow the convenient display of very long or complex SQL statements.

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

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

## Defining Informix CLI Output Data

When you read data from an Informix data source, the Informix CLI stage has an output link. The properties of this link and the column definitions of the data are defined on the **Output** page in the **InfmxCli Stage** dialog box.



The **Output** page has an **Output name** field and the **General**, **Columns**, **Selection**, and **SQL** tabs. The tabs displayed depend on how you specify the SQL statement to output the data. The **Columns...** and **View Data...** buttons function identically to those on the **Input** page.

- **Output name.** The name of the output link. Choose the link you want to edit from the **Output name** drop-down list box. This list box displays all the output links.
- Click **Columns...** to display a brief list of the columns designated on the input link. As you enter detailed meta data in the **Columns** tab, you can leave this list displayed.
- Click **View Data...** to invoke the Data Browser. This enables you to look at the data associated with the output link. For a description of the Data Browser, see DataStage documentation.

**General Tab.** This tab is displayed by default. It provides the interface for entering table names, the number of prefetch rows returned from Informix, and the type of query. The **General** tab contains the following fields:

- **Table names.** This field appears only when you select **Generated SQL query**. It contains the names of the Informix source tables or files being accessed. These tables must exist or be created and populated by Before SQL statements. There is no default.

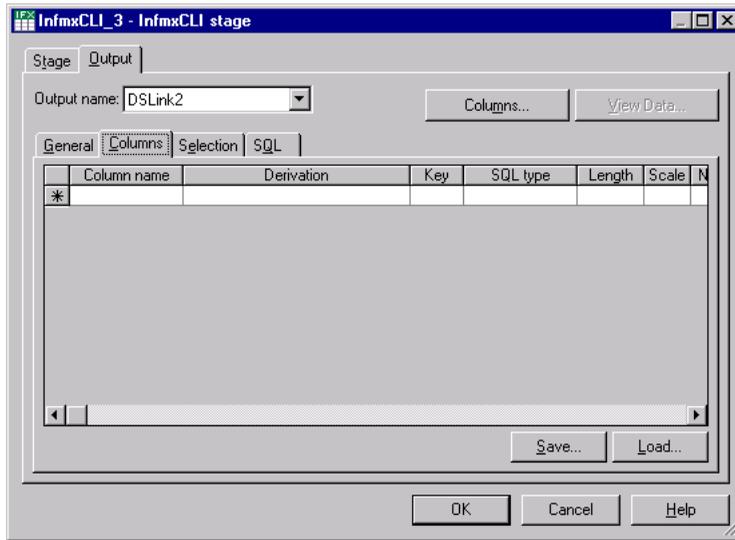
Separate multiple table names by a comma ( , ). You must have select privileges on each table. If you specify **User-defined SQL query**, **Table names** is ignored. You must specify a table name in **Table names** if you do not define **User-defined SQL query**.

Additionally, you can use a job parameter to specify the table name. For details on how to use define and use job parameters, see DataStage documentation.

You can also click the ... button at the right of the **Table name** field to browse the Repository to select the table.

- **Prefetch rows.** The number of rows that Informix returns when DataStage fetches data from the source tables. Specifying a value greater than 1 improves performance (memory usage increases to accommodate buffering multiple rows).
- **Query type.** Displays the **Generated SQL query** and **User-defined SQL query** options. **Query type** combines the “Generate SQL” and “User Defined SQL” grid properties.
  - **Generated SQL query.** This is the default setting, which specifies that the data is extracted using an SQL statement constructed by DataStage. When this option is selected, the **Generated** tab appears. You cannot edit this statement.
  - **User-defined SQL query.** Specifies that the data is extracted using a user-defined SQL query. When this option is selected, the **User-defined** tab appears allowing you to edit SQL statements.
- **Description.** Contains an optional description of the output link.

**Columns Tab.** This tab contains the column definitions for the data being output on the chosen link.

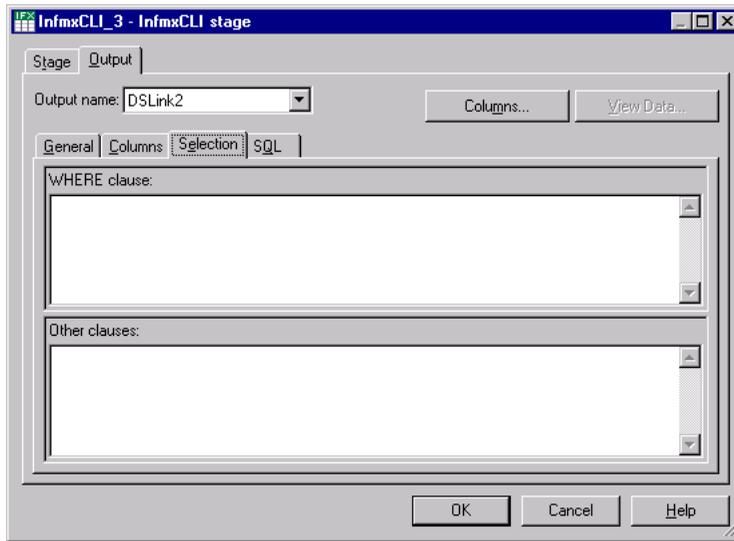


For a description of how to enter and edit column definitions, see DataStage documentation.

**Note:** The **Derivation** field on the **Columns** tab contains a fully qualified representation of column names when table definitions are loaded from the Repository.

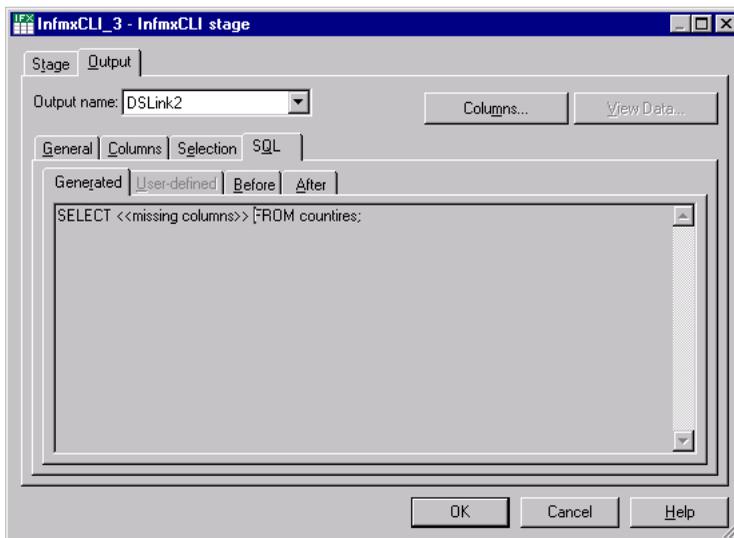
The column definitions for reference links require a key field. Key fields join reference inputs to a Transformer stage. The Informix CLI stage key reads the data by using a WHERE clause in the SQL SELECT statement.

**Selection Tab.** This tab is used primarily with generated SQL queries



It contains optional SQL SELECT clauses for the conditional extraction of data. These clauses are appended to the generated SQL statements.

**SQL Tab.** This tab displays the SQL statements (stage-generated or user-defined) or stored procedure call syntax used to read data from Informix.



It contains the **Generated**, **User-defined**, **Before**, and **After** tabs. These tabs are identical to those under the **SQL** tab for the **Input** page except there are no **Generated DDL** and **User-defined DDL** pages.

- **Generated.** This tab is displayed by default. It contains the SQL statements constructed by DataStage as a result of the **Update action** from the **General** tab of the **Output** page. You cannot edit these statements, but you can use **Copy** to copy them to the Clipboard for use elsewhere.
- **User-defined.** This tab contains the SQL statements executed to read data from Informix. This tab is enabled when you select **User-defined SQL query** or **User-defined SQL query file** from the **Query type** field on the **General** tab. The GUI displays the stage-generated SQL statement on this tab as a starting point. However, you can enter any valid, appropriate SQL statement. The box size changes proportionately when you resize the main window to display long SQL statements.
- **Before.** This tab contains the SQL statements executed before the stage processes any job data rows.
- **After.** This tab contains the SQL statements executed after the stage processes all job data rows.

The following sections describe the differences when you use SQL SELECT statements for generated queries or user-defined queries that you define on the **Output** page in the **InfmtxCLI Stage** dialog box of the GUI.

## Reading Data from Informix

The following sections describe the differences when you use generated queries or user-defined queries to read data from an Informix database into DataStage.

### Using Generated Queries

By default, DataStage extracts data from an Informix data source using an SQL SELECT statement that it constructs. The SQL statement is automatically constructed using the table and column definitions that you entered in the stage output properties.

When you select **Generated SQL query**, data is extracted from an Informix database using an SQL SELECT statement constructed by DataStage. 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 Informix CLI 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 **Name**, **Address**, and **Phone** columns 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 DataStage. However, you can use any of these SQL SELECT clauses:

SELECT clause	Specifies the columns to select from the database.
FROM clause	Specifies the tables containing the selected columns.
WHERE clause	Specifies the criteria that rows must meet to be selected.
GROUP BY clause	Groups rows to summarize results.
HAVING clause	Specifies the criteria that grouped rows must meet to be selected.
ORDER BY clause	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. These clauses are appended to the SQL statement that is generated by the stage. If this link is a reference link, only the WHERE clause is enabled.

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

- **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 GROUP BY, HAVING, or ORDER BY clause.

For more information about these clauses, see DataStage documentation.

## Using User-Defined Queries

Instead of using the SQL statement constructed by DataStage, you can enter your own SQL statement for each output link.

1. Select **User-defined SQL query** from the **Query type** drop-down list box on the **General** tab of the **Output** page. The **User-defined** tab on the **SQL** tab is enabled.
2. You can edit the statements or drag and drop 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.

If your entry begins with {FILE}, the remaining text is interpreted as a path-name, and the contents of the file supplies the text for the query.

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

## Mapping Data Types

You can map DataStage data types to Informix data types. When “Create Table” is set to Yes for input links, the target table is created using the input link column definitions and the specific input link properties that define the properties for the target table.

There is no exact translation between an Informix data type and a DataStage data type such as MONEY. The following table shows the Informix data types that are generated from the corresponding DataStage data types:

<b>DataStage Data Type</b>	<b>Informix Data Type</b>
SQL_BIGINT	INT8
SQL_BINARY	BYTE (Not supported)
SQL_BIT	BOOLEAN (Supported only with Informix Universal Server 9.n)
SQL_CHAR	CHAR( <i>n</i> )
SQL_DATE	DATE
SQL_DECIMAL	DECIMAL( <i>p, s</i> )

<b>DataStage Data Type</b>	<b>Informix Data Type</b>
SQL_DOUBLE	DOUBLE PRECISION
SQL_FLOAT	FLOAT( <i>n</i> )
SQL_INTEGER	INTEGER
SQL_LONGVARBINARY	BYTE (Not supported)
SQL_LONGVARCHAR	TEXT
SQL_NUMERIC	DECIMAL
SQL_REAL	REAL
SQL_SMALLINT	SMALLINT
SQL_TIME	DATETIME hour to second
SQL_TIMESTAMP	DATETIME
SQL_TINYINT	Not supported
SQL_VARBINARY	BYTE (Not supported)
SQL_VARCHAR	VARCHAR( <i>m,r</i> )

## Stored Procedure Support

You can call stored procedures from the server Informix CLI stage. The following restrictions apply:

- Output parameters are not supported.
- You can call stored procedures as part of the Before SQL and After SQL statements. Any result sets generated by the procedure are discarded.
- You can also call stored procedures as part of the “User Defined SQL.” For input links, the stored procedure must have at least the same number of input parameters as stage input columns, and the data type for the column should match the parameter type for the procedure. If there are more input parameters than columns, they must be specified as literals or job parameters.
- For output and reference links, the stored procedure must generate a row result set that matches the stage output column definitions. Additionally, a stored procedure for a reference link must produce one row based on the value of the primary key columns for the link.