

IBM InfoSphere DataStage and QualityStage
Version 11 Release 3

*Connectivity Guide for IBM UniVerse
and UniData*



IBM InfoSphere DataStage and QualityStage
Version 11 Release 3

*Connectivity Guide for IBM UniVerse
and UniData*



Note

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 37.

Contents

Chapter 1. Connecting to UniVerse and UniData databases 1

UniVerse File and Table Access	1
Data Sources Local to the InfoSphere DataStage Server	2
Data Sources Remote from the WebSphere DataStage Server	2
Co-resident Data Sources	3
UniData File Access	3
Accessing UniData	3
UniData File Meta Data.	4

Chapter 2. UniData stage 5

Using UniData Stages	5
Must Do's	5
Defining the Connection	5
UniData Connection Parameters	6
Defining Character Set Maps	6
Defining UniData Input Data	6
Defining UniData Output Data	7

Chapter 3. UniVerse stage 9

Using UniVerse Stages	9
Must Do's	9
Defining the Connection	10
UniVerse Connection Parameters	10
Defining UniVerse Input Data	11
Specifying Transaction Control Information.	12
Using a Generated Query.	14
Using a User-Defined SQL Statement.	14
Create Table Options	15
Defining UniVerse Output Data	15

Key Fields.	17
Using a Generated Query.	17
Using a User-Defined SQL Statement.	20

Chapter 4. Multivalued Data and Multilevel Files 21

Multivalued/Multi-Subvalued Data and Associations	21
Multiple Data Files (Multilevel Files)	23

Appendix A. Product accessibility 25

Appendix B. Reading command-line syntax 27

Appendix C. How to read syntax diagrams 29

Appendix D. Contacting IBM 31

Appendix E. Accessing the product documentation 33

Appendix F. Providing feedback on the product documentation 35

Notices and trademarks 37

Index 43

Chapter 1. Connecting to UniVerse and UniData databases

IBM® InfoSphere® DataStage® allows you to read data from or write data to UniVerse and UniData databases.

This chapter describes how to connect to UniVerse and UniData databases from a InfoSphere DataStage server job.

When you install InfoSphere DataStage, the UniData, and UniVerse stages are installed automatically and appear in the **Database** group on the server job palette.

UniVerse File and Table Access

You can access UniVerse data sources that are local to the InfoSphere DataStage server, as well as those on a remote server.

To access files in UniVerse systems from InfoSphere DataStage, you must define the systems in the *uvodbc.config* file in the relevant InfoSphere DataStage project directory on the server. The *uvodbc.config* file is a text file containing comments that describe how to add additional entries. The default *uvodbc.config* file looks like this:

```
*** To get to any ODBC source other than UniVerse, you need entries
*** that look as follows (the data source must also be configured
*** via the operating system's own mechanisms):
***
*** <data source name>
*** DBMSTYPE = ODBC
***
*** The local DataStage Server Engine is available via the data
*** source name "localuv" as defined below - please do not alter
*** this entry!
***
*** To access a remote UniVerse database, you need another entry
*** similar to that for localuv but with a remote host name in
*** place of "localhost".
***
*** To access a (coresident) UniVerse on the local machine, you
*** need to specify your local machine name or IP address in place
*** of "localhost".
***
*** Note that the spaces around the " = " signs are required, and
*** the data source name must be enclosed in angle brackets "<>".
***
[ODBC DATA SOURCES]
<localuv>
DBMSTYPE = UNIVERSE
network = TCP/IP
service = uvserver
host = localhost
```

You must not edit or delete the default InfoSphere DataStage entry in the file. To add an entry for a remote UniVerse system, follow the steps in “Data Sources Remote from the WebSphere® DataStage Server” on page 2. To access a co-resident UniVerse on the local machine, see “Co-resident Data Sources” on page 3.

Data Sources Local to the InfoSphere DataStage Server

The UniVerse stage provides access to tables in the local SQL catalog and hashed files in the account specified by **Account name**. Omitting **Account name** defaults to the current InfoSphere DataStage project.

The Hashed File stage provides access to hashed files. These are identified by **File name** and either **Account name** or **Account path**. The file name uniquely identifies a VOC record of type F or type Q in the specified account.

Data Sources Remote from the WebSphere® DataStage Server

The Hashed File stage provides access to UniVerse files on a remote UniVerse server that has UV/Net installed.

A remote account can be accessed by **Account name** after creating an entry in the UV.ACCOUNT file of the form:

```
ID: REMOTE 008 Remote ACCOUNT Pointer 011 mk9!/export/home/paton
```

The record ID is the account name, field 8 can optionally contain descriptive text, and field 11 contains the UniVerse host name and the path name of the account on the server system separated by a ! character.

Using UV/Net to access remote UniVerse files causes some confusion in the normally accepted client/server model. This is because, in general, the InfoSphere DataStage server itself becomes the client when accessing meta data over the UV/Net connection. The server is then the machine on which UniVerse is being accessed.

Connectivity is further complicated because the initial InfoSphere DataStage client/server link can be made over either LAN Manager or TCP/IP:

- To use LAN Manager, select the **Omit** check box in the Attach to Project dialog box. The same user name and password for InfoSphere DataStage client login must also work for the remote UniVerse server.
- To use TCP/IP, type your user name and password in the Attach to Project dialog box. Set the value for the UVNETRID environment variable as follows:

```
UVNETRID=HostName:PortNumbersUserNamesPassword
```

where s is the subvalue mark

Attach Mode	InfoSphere DataStage Server	UV/Net Server	Comment
User name and password	Windows	Windows	Import possible (see note 1)
		UNIX	Import possible
	UNIX	Windows	Import not possible (see note 2)
		UNIX	Import possible
Omit option selected	Windows	Windows	Import not possible (see note 2)
		UNIX	Import possible
	UNIX	Windows	No connection possible (expected)
		UNIX	No connection possible (expected)

Notes:

1. For the import to succeed, it is necessary to specify the host name in the format *hostname:LAN* in the UV.ACCOUNT record.
2. Connection over UV/Net is possible from the InfoSphere DataStage server prompt, but a remote ID must be set up. This can be done using the SET.REMOTE.ID command or by setting a value in the UVNETRID environment variable. The latter approach works for meta data import, but the former does not as the SET.REMOTE.ID command only sets the ID for the current UniVerse shell process.

The UniVerse stage provides access to UniVerse tables in the SQL catalog of a remote UniVerse server and UniVerse files in an account on that system specified by **Account name**. An account path can also be used. Before you can do this, you must create an entry in the *uvodbc.config* file in the InfoSphere DataStage project, which identifies the remote UniVerse server.

To add an entry for a remote UniVerse system:

1. Position your cursor at the bottom of the file and add the following lines, with the italicized variables changed to suit your system:

```
<name>  
DBMSTYPE = UNIVERSE  
network = TCP/IP  
service = uvserver  
host = hostname:portnumber
```

- *name* identifies the remote UniVerse system in any way you find useful and must be enclosed in angle brackets, as shown.
 - *hostname* specifies the host where UniVerse is installed and must be recognized as a node name by your TCP/IP system.
 - *Portnumber* specifies the port number for the connection. This is 31438 by default on UniVerse systems.
2. Save the file when you have added all the host names you require.

Co-resident Data Sources

Accessing tables and files on a UniVerse system that resides on the same machine as the InfoSphere DataStage server is the same as accessing remote UniVerse data sources.

Specify the local hostname as the host in the *uvodbc.config* file entry.

UniData File Access

You can concurrently access the same UniData server with the current version of InfoSphere DataStage and a previous version of InfoSphere DataStage

This is done by using different DSHELPER routines. Multiple versions of DSHELPER could exist on the UniData server over time, therefore you should remove older versions as needed to maintain the catalog space.

Accessing UniData

You can access UniData with the current version of InfoSphere DataStage and a previous version of InfoSphere DataStage by setting up the required environment variables.

Environment Variables on UNIX Systems

The UniData server requires that the environment variables UDTHOME and UDTBIN are set up on UNIX systems.

You should set these in your *.login* or *.profile* file. Assuming the default names from a basic installation, the variables are set as follows:

- From the Bourne or Korn shell:

```
UDTHOME=/usr/ud41;export UDTHOME
UDTBIN=/usr/ud41/bin;export UDTBIN
```
- From the C shell:

```
setenv UDTHOME /usr/ud41
setenv UDTBIN /usr/ud41/bin
```

Handling Large Records

You need to change the records per chunk value in order to accommodate very large records to UniData.

This is done by adding the following text in the **Description** box of the UniData stage Inputs page **General** tab at job design time:

```
RecsPerChunk= nnn
```

where *nnn* is the number of records returned per chunk.

You might need to use trial and error in getting the right value for this. You will get an error message when you run the job if you are attempting to return too many records per chunk.

UniData File Meta Data

When you import meta data from UniData databases, you select **Import > Table Definitions > UniData File Definitions** from the InfoSphere DataStage and QualityStage® Designer menu.

About this task

The resulting **Import Meta Data (UniData Files)** dialog box establishes a connection to a UniData database that is specified by a host name and a database name. When connected, a list of accessible files is displayed that includes all F- or LF-type file pointer records in the VOC that are not recognized as UniData system files.

When a file from this list is selected, the set of visible columns is determined solely from the contents of the file dictionary as follows:

- All D-, I-, and V-type dictionary items are shown including synonyms.

Chapter 2. UniData stage

The UniData stage represents a UniData file. This stage can be used to connect to UniData version 7.0 or earlier.

The UniData 6 stage is no longer supported. To connect to a UniData database, use the UniData stage, which offers better functionality and performance than the UniData 6 stage.

Using UniData Stages

You can use a UniData stage to extract or write data, or to act as an intermediate file in a job. Each UniData stage can have any number of inputs or outputs.

When you edit a UniData stage, the UniData Stage dialog box opens. This dialog box can have up to three pages (depending on whether there are inputs to and outputs from the stage):

- **Stage.** Displays the name of the stage you are editing. The **General** tab defines the data source name. You can add text to describe the purpose of the stage in the **Description** field. The **NLS** tab defines a character set map to use with the stage if NLS is enabled.
- **Inputs.** This page is displayed only if you have an input link to the stage. Specifies the data file to use and the associated column definitions for each data input link. This page also specifies how data is written to the data file.
- **Outputs.** This page is displayed only if you have an output link to the stage. Specifies the data file to use and the associated column definitions for each data output link.

Must Do's

You must define the data on the input and output links in the **UniData Stage** dialog box.

About this task

To edit a UniData stage, you carry out the following steps:

Procedure

1. Define the connection.
2. Define the data on the input links.
3. Define the data on the output links.

Results

These steps are performed in the **UniData Stage** dialog box. Click **OK** to close this dialog box. Changes are saved when you save the job design.

Defining the Connection

Before InfoSphere DataStage can access a UniData database, you must set values for the appropriate environment variables.

About this task

For details, see Chapter 1, “Connecting to UniVerse and UniData databases,” on page 1.

UniData Connection Parameters

To connect to a UniData data source, you must set the UniData connection parameters on the **General** tab.

Procedure

1. Choose a server name from the **Server** list or type a server name. The list is populated from the **Table Definitions > UniData** folder in the repository.
2. Specify the name of the database that you want to connect to.
3. Type the user name for logging on to the server in the **User name** field.
4. Type the password to use in the **Password** field.
5. Select the **UniVerse Stage Compatibility** check box to ensure that any job conversions will work correctly. With this option selected, the date or time will be represented in ISO format (depending on the Extended type) and numerics will be scaled according to the meta data. (The job conversion utility is a special standalone tool - it is not available within the Designer client.)

Results

You can use a job parameter in any of the fields described in steps 1 through 4.

Defining Character Set Maps

You can define a character set map for a UniData stage using the **NLS** tab in the **UniData Stage** dialog box.

The default character set map (defined for the project or the job) can be changed by selecting a map name from the list. The tab also has the following fields:

- **Show all maps.** Lists all the maps supplied with InfoSphere DataStage. Maps cannot be used unless they have been loaded using the InfoSphere DataStage and QualityStage Administrator.
- **Loaded maps only.** Display 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 UniData Input Data

When you write data to a UniData file, the UniData stage has an input link.

The properties of this link and the column definitions of the data are defined on the Inputs page in the **UniData Stage** dialog box.

The Inputs page has the following field and two 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 UniData stage.
- **General.** Displayed by default. Contains the following fields and options:
 - **File name.** The name of the file the data is written to. You can choose the file from the **File name** list, type a file name, or use a job parameter to represent

the file created during run time. This list contains all the files defined under the **Table Definitions > UniData > Server > Database** folder in the repository.

If the file you want is not listed, you can define a new table definition.

- **Clear file before writing.** If you select this check box, the existing file is cleared and new data records are written to the empty file. This check box is cleared by default.
- **Description.** Contains an optional description of the input link.

Note: If your data has very large records, you after need to change the records per chunk value to accommodate them. This is done by adding text in the **Description** box. See “Handling Large Records” on page 4 for more information.

- **Columns.** Contains the column definitions for the data written to the file.

Note: If you are using meta data that does not specify column positions, then the first column definition describes the UniData file's key field. The remaining columns are ordered according to their position in the file. Do *not* reorder the column definitions in the grid unless you are certain you understand the consequences of your action.

Click **View Data** to open the Data Browser. This enables you to look at the data associated with the input link.

Defining UniData Output Data

When you extract data from a UniData file, the UniData stage has an output link.

The properties of this link and the column definitions of the data are defined on the Outputs page in the **UniData Stage** dialog box.

The Outputs page has the following two fields and three tabs:

- **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 UniData stage.
- **Normalize on.** This list allows you to normalize (or unnest) data. You can normalize either on an association or on a single unassociated multivalued column. The **Normalize on** list is only enabled for nonreference output links where meta data has been defined that contains multivalued fields.
- **General.** Displayed by default. Contains the following fields and options:
 - **File name.** The name of the file the data is read from. You can use a job parameter to represent the file created during run time or choose the file from the **File name** list. This list contains all the files defined under the **Table Definitions > UniData > Server > Database** folder in the repository.
 - **Pre-load file to memory.** You can enable or disable this option using the box. Pre-loading a file into memory can improve performance if the output link is a reference input to a Transformer stage. If enabled, the UniData file is read into memory when the job is run. The default is **Disabled**.
 - **Description.** Contains an optional description of the output link.
- **Columns.** Contains the column definitions for the data on the chosen output link.

Note: If you are using meta data that does not specify column positions, then the first column definition describes the UniData file's key field. The remaining columns are ordered according to their position in the file. Do *not* reorder the column definitions in the grid unless you are certain you understand the consequences of your action.

- **Selection.** Contains optional SQL SELECT clauses for the conditional extraction of data from a file.

Click **View Data** to open the Data Browser. This enables you to look at the data associated with the output link.

Chapter 3. UniVerse stage

You can use the UniVerse stage to connect to UniVerse tables by using SQL.

Using UniVerse Stages

You can use a UniVerse stage to extract, write, or aggregate data.

You can use a Hashed File stage to connect to UniVerse files. Each UniVerse stage can have any number of inputs or outputs. Input links specify the data you are writing. Output links specify the data you are extracting and any aggregations required.

You can specify the data on an input or output link using an SQL statement constructed by InfoSphere DataStage or a user-defined query.

When you edit a UniVerse stage, the UniVerse Stage dialog box opens. This dialog box can have up to three pages (depending on whether there are inputs to and outputs from the stage):

- **Stage.** Displays the name of the stage you are editing. The **General** tab defines the data source name. You can describe the purpose of the stage in the **Description** field. The **NLS** tab appears only if you are connecting to a remote UniVerse database that is *not* NLS enabled.
- **Inputs.** This page is displayed only if you have an input link to this stage. Specifies the UniVerse 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 the data.
- **Outputs.** This page is displayed only if you have an output link to this stage. Specifies the UniVerse tables to use and the associated column definitions for each data output link. This page also contains the SQL SELECT statement or call syntax used to extract the data.

Must Do's

You must define the data on the input and output links in the **UniVerse Stage** dialog box.

About this task

To edit a UniVerse stage, you carry out the following steps:

Procedure

1. Define the connection.
2. Define the data on the input links.
3. Define the data on the output links.

Results

These steps are performed in the **UniVerse Stage** dialog box. Click **OK** to close this dialog box. Changes are saved when you save the job design.

Defining the Connection

To connect to a UniVerse data source, the data source must be configured in the *uvodbc.config* file.

About this task

For more information, see Chapter 1, “Connecting to UniVerse and UniData databases,” on page 1.

The InfoSphere DataStage Engine can host UniVerse tables. Such tables are accessed by the data source name *localuv*, which is already set up in the *uvodbc.config* file.

UniVerse Connection Parameters

The UniVerse connection parameters are set on the **General** and **Details** tabs on the Stage page.

About this task

Procedure

To connect to a UniVerse data source:

1. Choose the data source name from the **Data source name** list. This list contains all the data sources defined in the **Table Definitions > UniVerse** folder in the repository.
If the data source name you want is not listed, you can either type the name in the **Data source name** field or create a table definition.
2. Type the user name to use in the **User name** field. You can specify a job parameter here, in the form *#parameter#*.
3. Type the password to use in the **Password** field. You can specify a job parameter here, in the form *#parameter#*, but it will be displayed as asterisks.

Note: If you are connecting to a remote UniVerse system using LAN Manager, or accessing a data file in *localuv*, you do not need to specify a user name and password. In the case of *localuv* data sources, the **User name** and **Password** fields are disabled.

4. Type an optional description of the UniVerse stage in the **Description** field.
5. If you are using a DSN other than *localuv*, click the **Details** tab. The **Details** tab appears at the front of the Stage page.
6. Specify the UniVerse account to use for the connection either by account name or account directory:
 - **Use account name.** Type the account name in the **Account name** field. This must represent a valid UniVerse account which is also a schema.
 - **Use directory path.** Specify the directory path where the project is stored in the **Directory path** field.

Note: If you are accessing a file on *localuv*, you do not need to specify an account or directory path, and the **Details** tab is disabled.

Results

You can use a job parameter in any of the fields described in steps 1, 2, 3, 4, and 6.

Defining UniVerse Input Data

When you write data to a table in a UniVerse database, the UniVerse stage has an input link.

The properties of this link and the column definitions of the data are defined on the Inputs page of the UniVerse Stage dialog box.

The Inputs page has the following field and up to six tabs, depending on the **Update action** you select in the **General** tab, and whether you want to create a table in the target database:

- **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 UniVerse stage.
- **General.** This tab is displayed by default. It contains the following parameters:
 - **Table name.** The name of the table or UniVerse file the data is written to. Choose the table or file from the **Table name** list. This list contains all the tables defined in the **Table Definitions > UniVerse > Data source** folder in the repository. *Data source* is the data source name chosen on the **General** tab on the Stage page.

If the table you want is not listed, you need to define a table definition. Alternatively, use **Browse...** to display the Table Definitions dialog box and choose a suitable table definition.

You can also specify a job parameter in this field.

This field appears when the update action is *not* **User-defined SQL**.
 - **Update action.** Specifies how the data is written. Choose the option you want from the list:
 - Clear the table, then insert rows.** Deletes the contents of the table and adds the new rows.
 - Insert rows without clearing.** Inserts the new rows in the table.
 - Insert new or update existing rows.** New rows are added or, if the insert fails, the existing rows are updated.
 - 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. If a row with the supplied key does not exist in the table, then the table is not updated and a warning is logged.
 - Update existing rows or insert new ones.** The existing data rows are updated and new rows are added.
 - User-defined SQL.** The data is written using a user-defined SQL statement. When you select this option, the **View SQL** tab is replaced by the **Enter SQL** tab.
 - **Create table in target database.** Select this check box if you want to automatically create a table in the target database at run time. A table is created based on the defined column set for this stage. If you select this box, an additional tab, **Edit DDL**, appears. This shows the SQL CREATE statement to be used for table generation.
 - **Description.** Contains an optional description of the input link.
- **Columns.** This tab is always present and 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.

- **View SQL.** This tab displays the SQL statement call syntax used to write the data. This tab appears when you select any update action other than **User-defined SQL**. You cannot edit this statement, but you can use **Copy** to copy it to the Clipboard for use elsewhere.
- **Enter SQL.** This tab displays the user-defined SQL statement. This tab appears only when you set the update action to **User-defined SQL**, when it replaces the **View SQL** tab.
- **Edit DDL.** This tab appears if you have chosen to automatically generate a table at run time by selecting the **Create table in target database** check box on the **General** tab. It displays the SQL CREATE statement that will be used to create the table. To generate the statement, click **Create DDL**. InfoSphere DataStage will connect to the target database and generate the statement. You can edit the statement on this tab to make any required changes. This tab also allows you to specify that any existing table by this name should be dropped first. If you do not select this option, and such a table already exists in the target database, then the create will fail. You can specify details about the table to be created in the Create table options dialog box. Open this by clicking **Options**. For more information see “Create Table Options” on page 15.
- **Transaction Handling.** This tab allows you to specify the transaction handling features of the stage as it writes to the UniVerse data source. You can choose whether to use transaction grouping or not, specify an isolation level, and specify the number of rows written before data is committed. A grid shows details of the transaction group to which the currently selected input link belongs. For further information, see “Specifying Transaction Control Information.”

Click **View Data** to open the Data Browser. This enables you to look at the data associated with the input link.

Specifying Transaction Control Information

Multiple input links writing to a single UniVerse data source can be associated together as a transaction group. The transaction grouping feature is turned on and off using the **Enable transaction grouping** check box on the **Transaction Handling** tab (it is off by default).

About this task

If transaction grouping is off, you can specify the following information on the **Transaction Handling** tab:

- Type a suitable value in the **Rows per transaction** field. This is the number of rows written before the data is committed to the data table. The default value is 0, that is, all the rows are written before being committed to the data table.
- Select a suitable **Isolation Level**. The isolation level specifies how potential conflicts between transactions (for example, dirty reads, nonrepeatable reads, and phantom reads) are handled.

If transaction grouping is enabled, the following rules govern the grouping of links:

- All the input links in the transaction group must originate from the same Transformer stage.
- The ordering of the links within the transaction group is determined in the preceding Transformer stage.

- A transaction group cannot use a **Rows per transaction** other than 1. Using an **Isolation level** of **Auto-commit** is permitted, but obviates the effect of organizing links in a transaction group.

You should be aware of the following facts about transaction groups (assuming that you commit on every row):

- A transaction starts at the beginning of each iteration of the Transformer stage preceding the UniVerse stage. Any uncommitted changes left over from a previous transaction are rolled back.
- The links in the transaction group are processed in the order laid down in the Transformer stage. Individual links can be skipped if constraints laid down in the preceding Transformer stage so dictate.
- Each link in the transaction group can specify whether to rollback on failure. A rollback on any link causes the transaction to be abandoned and any subsequent links in the group to be skipped.
- Each link in the transaction group can be set to rollback if a constraint on that link is not met. Again, such a rollback causes the transaction to be abandoned and any subsequent links in the group to be skipped.
- The row counter for each link will be incremented only if the SQL associated with the link executes successfully and the transaction is successfully committed.
- The transaction ends after the last link in the transaction group is processed, unless a preceding link performs a rollback, in which case the transaction ends there.

To specify transaction control information for a transaction group:

Procedure

1. Click the **Transaction Handling** tab.
2. Select the **Enable transaction grouping** check box.
3. Choose an appropriate transaction isolation level to use from the **Isolation level** list. The isolation level specifies how potential conflicts between transactions (for example, dirty reads, nonrepeatable reads, and phantom reads) are handled. (If you select **Auto-commit**, you are specifying that every statement will effectively be executed in a separate transaction, which will obviate the advantages of transaction groups.)
4. For transaction groups, **Rows per transaction** is automatically set to 1 and you cannot alter it.
5. Supply necessary details about the transaction group in the grid. The grid has a line for every link in the transaction group. The links are shown in transaction processing order, which is set in the preceding Transformer stage. Each line contains the following information:
 - **Input name.** The name of the input link. You cannot change this.
 - **On Skip.** This is used to specify whether to continue or to roll back if a link is skipped due to a constraint on it not being satisfied. Choose **Continue** or **Rollback** from the list.
 - **On Fail.** This is used to specify whether or not to continue or rollback on failure of the SQL statement. Choose **Continue** or **Rollback** from the list.
 - **SQL.** Shows the SQL statement associated with the input link. You cannot change this, but clicking the cell will display the entire statement.

Note: If the UniVerse stage uses a local connection to DSEngine (the data source is *localuv*), then you cannot associate data written from different input

links in a single transaction group. Every input link appears in a transaction group on its own, even if the links originate from the same Transformer stage. You can view the transaction group information for a particular link by choosing it from the **Input name** list.

Using a Generated Query

About this task

You can write data to an SQL table using an SQL statement constructed by InfoSphere DataStage. When you specify the table and the column definitions to use, the SQL statement is automatically constructed and can be viewed on the **View SQL** tab.

To use a generated query:

Procedure

1. Choose a table from the **Table name** list on the **General** tab.
2. Specify how you want the data to be written by choosing a suitable option from the **Update action** list. There are six options for a generated query:
 - **Clear the table, then insert rows**
 - **Insert rows without clearing**
 - **Insert new or update existing rows**
 - **Replace existing rows completely**
 - **Update existing rows only**
 - **Update existing or insert new rows**See “Defining UniVerse Input Data” on page 11 for a description of each update action.
3. Type an optional description of the input link in the **Description** field.
4. Click the **Columns** tab. The **Columns** tab appears at the front of the Inputs page.
5. 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.
6. Click the **View SQL** tab. The **View SQL** tab appears at the front of the Inputs page.
7. Click **OK** to close the UniVerse Stage dialog box. Changes are saved when you save your job design.

Using a User-Defined SQL Statement

You can specify your own SQL statement for each UniVerse input link instead of writing data using an SQL statement constructed by InfoSphere DataStage.

Procedure

To specify an SQL statement:

1. Choose **User-defined SQL** from the **Update action** list on the **General** tab. The **View SQL** tab is replaced with the **Enter SQL** tab.
2. Click the **Columns** tab. The **Columns** tab appears at the front of the Inputs page.

3. Edit the Columns grid to specify column definitions for the columns you want to write.
4. Click the **Enter SQL** tab. The **Enter SQL** tab appears at the front of the Inputs page.
5. Type the SQL statement you want to use. This statement must contain the table name, the type of update action you want to perform, and the columns you want to write.
Verify that the statement contains the correct number of parameter markers. You must have a parameter marker for each column you have defined on the **Columns** tab.
6. Click **OK** to close the **UniVerse Stage** dialog box. Changes are saved when you save your job design.

Create Table Options

The Create table options dialog box allows you to specify various options about how the underlying file holding the table is created if you choose **Options** or **Create DDL** from the **Edit DDL** tab.

The dialog box contains the following fields:

- **File type.** The file type chosen determines what other options are available in the dialog box. The default is Type30(Dynamic).
- **Minimum modulus.** Visible only for Type30(Dynamic) file types. Specifies the dynamic file minimum modulus in the range 1 to 999999. The default is 1.
- **Group size.** Visible only for Type30(Dynamic) file types. Specifies the dynamic group size. Choose 1 to select a group size of 2048 bytes, or 2 to select a group size of 4096 bytes. The default is 1.
- **Split load.** Visible only for Type30(Dynamic) file types. Specifies the dynamic file split as a percentage in the range 1 to 99. The default is 80.
- **Merge load.** Visible only for Type30(Dynamic) file types. Specifies the dynamic file merge load as a percentage in the range 1 to 99. The default is 50.
- **Large record.** Visible only for Type30(Dynamic) file types. Specifies the large record value in bytes in the range 1 to 999999. The default is 80.
- **Hash algorithm.** Visible only for Type30(Dynamic) file types. Specifies the dynamic file hashing algorithm. Choose from **GENERAL** or **SEQ.NUM**. The default is **GENERAL**.
- **Record size.** Visible only for Type30(Dynamic) file types. Specifies the record size in the range 1 to 999999.
- **Minimize space.** Visible only for Type30(Dynamic) file types. Select this to specify that some of the other options are adjusted to optimize for minimum file size.
- **Modulus.** Visible only for hashed file types. Specifies the hashed file modulus in the range 1 to 999999. The default is 1.
- **Separation.** Visible only for hashed file types. Specifies the hashed file separation in the range 1 to 999999. The default is 2.

Defining UniVerse Output Data

When you extract data from a UniVerse data source, the UniVerse stage has an output link.

The properties of this link and the column definitions of the data are defined on the Outputs page in the **UniVerse Stage** dialog box.

The Outputs page has the following two fields and up to five tabs, depending 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 UniVerse stage.
- **Normalize on.** This list allows you to normalize (or unnest) data. You can normalize either on an association or on a single unassociated multivalued column. The **Normalize on** list is only enabled for nonreference output links where meta data has been defined that contains multivalued fields.
- **General.** Displayed by default. Contains the following components:
 - **Table names.** Contains the names of the tables or files being accessed in a comma-separated list. You can also use a job parameter to specify the table name.
 - **Available tables.** Displays the names of the available tables or files that have definitions in the repository.
 - **Add.** Adds a table from the **Available tables** list to the **Table names** field.
 - **Generated query.** Specifies that the data is extracted using an SQL statement constructed by InfoSphere DataStage. This is the default setting. When this option is selected, the **Selection** and **View SQL** tabs appear.
 - **User-defined SQL query.** Specifies that the data is extracted using a user-defined SQL query. When this option is selected, the **SQL Query** tab appears.
 - **Description.** Contains an optional description of the output link.
 - **Browse .** Displays the Table Definitions dialog box, allowing you to choose a suitable table or stored procedure definition.
- **Columns.** Contains the column definitions for the data being output on the chosen link. Also specifies which columns are aggregated.
- **Selection.** Contains optional SQL SELECT clauses for the conditional extraction of data. This tab appears when you select **Generated query**.
- **View SQL.** Displays the SQL statement used to extract the data from the chosen table or tables. This tab appears when you select **Generated query**. The SQL statement exists in two forms and you can choose which one to display:
 - **SQL for reference inputs.** Choose this to view the SQL statement used when this link is a reference input to a Transformer stage.
 - **SQL for primary inputs.** Choose this to view the SQL statement used in all other cases.
You cannot edit the SQL statement, but you can use **Copy** to copy it to the Clipboard for use elsewhere.
- **SQL Query.** Contains a user-defined SQL query. This tab appears when you select **User-defined SQL query**. This tab is divided into two areas:
 - **SQL for primary inputs.** Contains a user-defined SQL query for a link that is a primary input to a Transformer stage, or an input to any other type of stage.
 - **SQL for reference inputs.** Contains a user-defined SQL query for a link that is a reference input to a Transformer stage.
- **Transaction Handling.** Allows you to specify a transaction isolation level for read data. The isolation level specifies how potential conflicts between transactions (for example, dirty reads, nonrepeatable reads, and phantom reads) are handled.

Click **View Data** to open the Data Browser. This enables you to look at the data associated with the input link.

Key Fields

Key fields are used to join primary and reference inputs to a Transformer stage.

The column definitions for output links contain a key field. For details about how key fields are specified and used, see *IBM InfoSphere DataStage Server Job Developer's Guide*.

Using a Generated Query

When you select **Generated query**, data is extracted from a UniVerse data source using an SQL SELECT statement constructed by InfoSphere 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 UniVerse stage, the SQL SELECT statement is automatically constructed and can be viewed by clicking the **View SQL** tab on the Outputs page.

Note: The **View SQL** tab appears only when you select **Generated query** on the **General** tab on the Outputs page.

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

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

The SELECT and FROM clauses are the minimum required and are automatically generated by InfoSphere 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. See "Aggregating Data" on page 19 for a description of how this clause is used.

HAVING clause

Specifies the criteria that grouped rows must meet to be selected. See "Aggregating Data" on page 19 for a description of how this clause is used.

ORDER BY clause

Sorts selected rows.

If you want to use the additional SQL SELECT clauses, you must specify them on the **Selection** tab on the Outputs page. The **Selection** tab is divided into two parts:

- **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 a WHERE Clause

You can use a WHERE clause to select only the data that meets certain criteria and join two tables from the same data source.

To use a WHERE clause, type the column and the condition in the **WHERE clause** text entry box.

For example, if you have a table (Sales1) containing sales data, you can choose to only output data where the value in the **Price** column is greater than \$10.00. In this case, specify:

```
Price>10
```

Alternatively, if you are extracting data from two tables in the data source, you can use a WHERE clause to relate a column in one table to a column in the another table.

For example, Table1 contains the columns **Pcode**, **OrderNo**, and **SaleDate** and Table2 contains **Pcode**, **CustNo**, **Quantity**, and **Cost**. You can use the WHERE clause to join the two tables together by the related column. In this case, the column is **Pcode** and you specify:

```
Table1.Pcode = Table2.Pcode
```

Note: Only one column definition called **Pcode** is loaded or inserted into the grid on the **Columns** tab.

You can also use a job parameter in the WHERE clause.

The SQL SELECT statement is automatically updated to include the WHERE clause. Click the **View SQL** tab to display the statement.

Using a HAVING Clause

You can use a HAVING clause to specify conditions that the grouped data must meet before it is selected if you use a UniVerse stage to aggregate data.

For more information about using a UniVerse stage to aggregate data, see “Aggregating Data” on page 19.

To use a HAVING clause, type the clause, column, and condition in the **Other clauses** text entry box on the **Selection** tab on the Outputs page.

For example, you might choose to only output summed quantities that are greater than or equal to 1000. In this case you specify:

```
HAVING SUM(QtySold)>=1000
```

The SQL SELECT statement is updated automatically. Click the **View SQL** tab to display the statement.

You can also use a job parameter in the HAVING clause.

Using an ORDER BY Clause

You can sort data based on a chosen column by including an ORDER BY clause in the SELECT statement.

Records are sorted by data in the chosen column before being output. You can specify a column name or a column position and whether to sort in ascending or descending order.

To use an ORDER BY clause, type the clause, column, and condition in the **Other clauses** text entry box on the **Selection** tab on the Outputs page.

For example, if your table contains a **Name** column, you might want to sort the column alphabetically (A to Z). In this case you specify:

```
ORDER BY Name ASC
```

The SQL SELECT statement is updated automatically. Click the **View SQL** tab to display the statement.

Aggregating Data

You can use a UniVerse stage to aggregate data at the source instead of using an intermediate Aggregator stage if you are using a generated query.

By aggregating data you can add values in a particular column for all data records in a table. This summed total is then output from the stage.

You can aggregate data in two ways:

- Using an Aggregator stage.
- Using a UniVerse stage.

If you aggregate data using a UniVerse stage, the columns to group by and sum together are also specified by the SQL SELECT statement. To specify the columns to group by and summarize, you must edit the column definitions in the Columns grid on the **Columns** tab.

For example, if you have a sales database (Sales1) it might contain the following columns: **Product**, **SaleDate**, and **QtySold**. If this database is updated daily, you have a record of how many of each product are sold each day. However, if you want to know how many of each product were sold since 01/01/96 you need to specify a WHERE clause for the **SaleDate** and group (and summarize) the data.

Because you want the total for each product, you need to group all the occurrences of the same value in the **Product** column and sum the value in the **QtySold** column.

To group by a column, click in the **Group** cell for the column definition you want to group by and choose **Yes** from the list. In the example, you would choose the **Product** column to edit.

To summarize a column, click in the **Derivation** cell for the column you want to aggregate (using SUM or COUNT). By default the **Derivation** cell contains the name of the table and column in the format *tablename.columnname*. You can edit this cell to add SUM or COUNT. In the example, you would edit the **Derivation** cell for the **QtySold** column. The resulting expression would be SUM(Sales1.QtySold).

You can use the Expression Substitution dialog box to edit multiple **Derivation** cells at the same time. Select the columns and choose **Derivation Substitution** from the shortcut menu.

When you group by or summarize columns, the SQL statement is automatically updated to include the GROUP BY clause and the aggregation expression. To view the SQL statement, click the **View SQL** tab on the Outputs page.

For example, the SQL statement for the example would be:

```
SELECT Product, SUM(QtySold) FROM Sales1
WHERE Saledate>=01/01/96
GROUP BY Product;
```

See “Using a WHERE Clause” on page 18 for details of how to use a WHERE clause.

Using a User-Defined SQL Statement

you can specify your own SQL statement for each UniVerse output link instead of using the SQL statement constructed by InfoSphere DataStage.

Procedure

To specify an SQL statement:

1. Click the **User-defined SQL query** option on the **General** tab on the Outputs page. The **SQL Query** tab appears.
2. Click the **SQL Query** tab. When you first view this tab, the **SQL for primary inputs** and **SQL for reference inputs** fields might contain the SQL statements constructed by InfoSphere DataStage. These are displayed if you selected **Generated query** before selecting **User-defined SQL query**. You can modify or overwrite each statement to construct your own SQL query or call to a stored procedure.

The entries in these fields depend on whether the output is a primary input to a stage or a reference input to a Transformer stage:

- If the output is a primary input to any stage, whether or not it is a Transformer stage, edit the **SQL for primary inputs** field. The SQL query must contain the same number of columns (and column names) as the SQL statement constructed by InfoSphere DataStage.

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.

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

- If the output is a reference input to a Transformer stage, edit the **SQL for reference inputs** field. The SQL query must contain the same number of columns as the SQL statement constructed by InfoSphere DataStage. You must ensure that the table definitions for the output link are correct and represent the columns that are expected. The statement must have the same number of parameter values (?) as key columns on the link. The result set generated by this statement or procedure contains at most one row.
3. Click **OK** to close the UniVerse Stage dialog box. Changes are saved when you save your job design.

Chapter 4. Multivalued Data and Multilevel Files

This appendix describes how meta data that is imported from UniVerse or UniData is determined to be multivalued and multi-subvalued, and the normalization options that are available in the IBM InfoSphere DataStage.

It also describes how multilevel files are handled in InfoSphere DataStage.

Multivalued/Multi-Subvalued Data and Associations

Both UniVerse and UniData support multivalued and multi-subvalued data and associations, which allow a relationship between a number of multivalued/multi-subvalued fields within a record to be identified.

For IBM InfoSphere DataStage 3.1 and later, meta data imported from UniVerse or UniData includes information about whether a field is single-valued, multivalued, or multi-subvalued, and whether it is part of an association. This is determined as follows in both UniVerse and UniData dictionaries:

- The value of field 6 indicates the field type. S or " denotes single-valued, M or MV denotes multivalued, and MS denotes multi-subvalued.
- The value of field 7 is the name of the association to which the field belongs.
- A field that is multivalued and does not belong to an association is treated as a member of an association whose name is the same name as the field.

UniVerse dictionaries can also contain Pick-style A- and S-type entries which are interpreted differently:

- The value of field 4 indicates a multivalued attribute. It contains an entry of the form **C;n[m]...** for a controlling attribute or **D;n** for a dependent attribute. When importing meta data, this is handled like an association with the same name as the controlling attribute. All the controlling and dependent attributes appear as multivalued fields that are members of this association.
- The value of field 5 can be used to override the field type. S or " denotes single-valued, M or MV denotes multivalued, and MS denotes multi-subvalued.

UniVerse, UniData, and Hashed File stage outputs recognize the presence of multivalued and multi-subvalued column meta data. These stages offer the user the option to normalize the data with respect to a single association generating a separate row of column values for each multivalued in an association. The Un-Normalized options deliver all of the values as a dynamic array containing value marks and subvalue marks.

The Data Browser **Display...** button also offers the Un-Normalized (formatted) option, which formats the data in the grid with each value on a separate line in a similar way to Retrieve or UniQuery.

The following example shows how records containing multivalued data are displayed for a table with six columns: two single-valued, two multivalued and two multi-subvalued. The multivalued and multi-subvalued fields all belong to a single association.

Column Name	Field No.	Field Type	Association
Key	0	S	
Field1	1	S	
Field2	2	MV	Assoc1
Field3	3	MV	Assoc1
Field4	4	MS	Assoc1
Field5	5	MS	Assoc1

The sample records contain the following data:

ID: A
 DATA: BfmCvmGvmNvmRfmDvmHvmOvmSfmEvmIsmKsmMvmPvmTsmVfmFvmJsmLvmQvmUsmW
 ID: X
 DATA: Yfmfmfmfm

The second record is used to illustrate the differences in normalization behavior between UniVerse and UniData when the associated multivalued fields are empty.

The unnormalized view of the table returns the following column values:

Key	Field1	Field2	Field3	Field4	Field5
A	B	CvmGvmNvmR	DvmHvmMvmS	EvmIsmKsmMvmPvmTsmV	FvmJsmLvmQvmUsmW
X	Y				

The unnormalized (formatted) view of the table, only available from the Data Browser, formats the previous row as follows. This is similar to the layout that Retrieve or UniQuery produce.

Key	Field1	Field2	Field3	Field4	Field5
A	B	C	D	E	F
		G	H	I	J
				K	L
				M	Null
		N	O	P	Q
		R	S	T	U
				V	W
X	Y	Null	Null	Null	Null

The view of the table normalized on Assoc1 will vary slightly between UniData files, UniVerse files, and UniVerse tables.

Key	Field1	Field2	Field3	Field4	Field5
A	B	C	D	E	F
A	B	G	H	I	J
A	B	G	H	K	L
A	B	G	H	M	Null

Key	Field1	Field2	Field3	Field4	Field5
A	B	N	O	P	Q
A	B	R	S	T	U
A	B	R	S	V	W
X	Y	Null	Null	Null	Null

The result set above will be returned by the UniData stage or the Data Browser during UniData file meta data import, and is equivalent to that returned by the UniData SQL query:

```
SELECT Key,Field1,Field2,Field3,Field4,Field5 FROM Table1
UNNEST Field2,Field3,Field4,Field5
```

The result set returned by the Hashed File stage or the Data Browser during UniVerse file meta data import differs from the above in that the final row will be omitted:

Key	Field1	Field2	Field3	Field4	Field5
A	B	C	D	E	F
A	B	G	H	IsmKsmM	JsmL
A	B	N	O	P	Q
A	B	R	S	TsmV	UsmW

The result set above will be returned by the UniVerse stage or the Data Browser during UniVerse table meta data import, and is equivalent to that returned by the UniVerse SQL query:

```
SELECT Key,Field1,Field2,Field3,Field4,Field5
FROM UNNEST Table1 ON Assoc1
```

Multiple Data Files (Multilevel Files)

Both UniVerse and UniData support Pick-style multiple data files (multilevel files), which allow multiple subfiles to share a common dictionary.

When importing meta data for UniVerse files or tables or UniData files, only a single file or table corresponding to the shared dictionary and default data subfile will be visible. This can be used to import the meta data from the shared dictionary in the same way as for any other files or tables.

In a InfoSphere DataStage job, a specific subfile can be used in a Hashed File or UniData stage input/output by specifying a file name of the form *MFileName,SubFileName*.

The default subfile can also be specified by a file name of the form *MFileName*. This form can also be used in a UniVerse stage input/output.

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.

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. Because the information center uses XHTML, you can set display preferences in your browser. This 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.

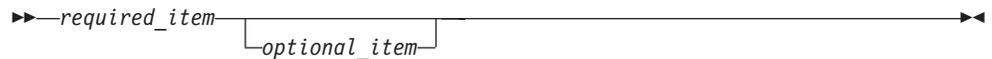
Appendix C. How to read syntax diagrams

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).



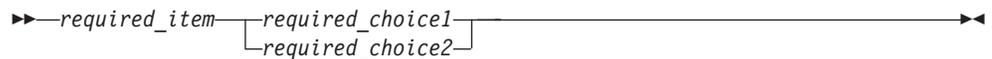
- Optional items appear below the main path.



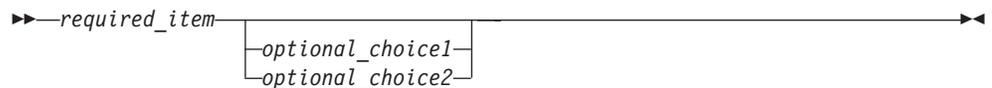
If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.



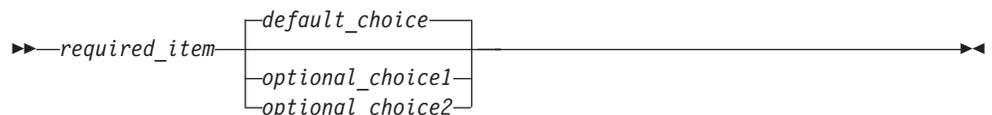
- If you can choose from two or more items, they appear vertically, in a stack. If you must choose one of the items, one item of the stack appears on the main path.



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



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



- 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 1. IBM resources

Resource	Description and location
IBM Support Portal	You can customize support information by choosing the products and the topics that interest you at www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server
Software services	You can find information about software, IT, and business consulting services, on the solutions site at www.ibm.com/businesssolutions/
My IBM	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 www.ibm.com/account/
Training and certification	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 http://www.ibm.com/training
IBM representatives	You can contact an IBM representative to learn about solutions at www.ibm.com/connect/ibm/us/en/

Appendix E. Accessing the product documentation

Documentation is provided in a variety of formats: in the online IBM Knowledge Center, in an optional locally installed information center, and as PDF books. You can access the online or locally installed help directly from the product client interfaces.

IBM Knowledge Center is the best place to find the most up-to-date information for InfoSphere Information Server. IBM Knowledge 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 IBM Knowledge Center from the installed product or from a web browser.

Accessing IBM Knowledge Center

There are various ways to access the online documentation:

- Click the **Help** link in the upper right of the client interface.
- 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.

- Type the address in a web browser, for example, when you are not logged in to the product.

Enter the following address to access all versions of InfoSphere Information Server documentation:

```
http://www.ibm.com/support/knowledgecenter/SSZJPZ/
```

If you want to access a particular topic, specify the version number with the product identifier, the documentation plug-in name, and the topic path in the URL. For example, the URL for the 11.3 version of this topic is as follows. (The ⇒ symbol indicates a line continuation):

```
http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/⇒  
com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html
```

Tip:

The knowledge center has a short URL as well:

```
http://ibm.biz/knowctr
```

To specify a short URL to a specific product page, version, or topic, use a hash character (#) between the short URL and the product identifier. For example, the short URL to all the InfoSphere Information Server documentation is the following URL:

```
http://ibm.biz/knowctr#SSZJPZ/
```

And, the short URL to the topic above to create a slightly shorter URL is the following URL (The ⇒ symbol indicates a line continuation):

```
http://ibm.biz/knowctr#SSZJPZ_11.3.0/com.ibm.swg.im.iis.common.doc/⇒  
common/accessingiidoc.html
```

Changing help links to refer to locally installed documentation

IBM Knowledge Center contains the most up-to-date version of the documentation. However, you can install a local version of the documentation as an information center and configure your help links to point to it. A local information center is useful if your enterprise does not provide access to the internet.

Use the installation instructions that come with the information center installation package to install it on the computer of your choice. After you install and start the information center, you can use the **iisAdmin** command on the services tier computer to change the documentation location that the product F1 and help links refer to. (The `⇒` symbol indicates a line continuation):

Windows

```
IS_install_path\ASBServer\bin\iisAdmin.bat -set -key ⇒  
com.ibm.iis.infocenter.url -value http://<host>:<port>/help/topic/
```

AIX® Linux

```
IS_install_path/ASBServer/bin/iisAdmin.sh -set -key ⇒  
com.ibm.iis.infocenter.url -value http://<host>:<port>/help/topic/
```

Where `<host>` is the name of the computer where the information center is installed and `<port>` is the port number for the information center. The default port number is 8888. For example, on a computer named `server1.example.com` that uses the default port, the URL value would be `http://server1.example.com:8888/help/topic/`.

Obtaining PDF and hardcopy documentation

- The 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>.

Appendix F. Providing feedback on the product documentation

You can provide helpful feedback regarding IBM documentation.

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

- To provide a comment about a topic in IBM Knowledge Center that is hosted on the IBM website, sign in and add a comment by clicking **Add Comment** button at the bottom of the topic. Comments submitted this way are viewable by the public.
- To send a comment about the topic in IBM Knowledge Center to IBM that is not viewable by anyone else, sign in and click the **Feedback** link at the bottom of IBM Knowledge 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).

Notices and trademarks

This information was developed for products and services offered in the U.S.A. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session or persistent cookies. If a product or component is not listed, that product or component does not use cookies.

Table 2. Use of cookies by InfoSphere Information Server products and components

Product module	Component or feature	Type of cookie that is used	Collect this data	Purpose of data	Disabling the cookies
Any (part of InfoSphere Information Server installation)	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
Any (part of InfoSphere Information Server installation)	InfoSphere Metadata Asset Manager	<ul style="list-style-type: none"> • Session • Persistent 	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication • Enhanced user usability • Single sign-on configuration 	Cannot be disabled

Table 2. Use of cookies by InfoSphere Information Server products and components (continued)

Product module	Component or feature	Type of cookie that is used	Collect this data	Purpose of data	Disabling the cookies
InfoSphere DataStage	Big Data File stage	<ul style="list-style-type: none"> • Session • Persistent 	<ul style="list-style-type: none"> • User name • Digital signature • Session ID 	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere DataStage	XML stage	Session	Internal identifiers	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere DataStage	IBM InfoSphere DataStage and QualityStage Operations Console	Session	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Data Click	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Data Quality Console		Session	No personally identifiable information	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere QualityStage Standardization Rules Designer	InfoSphere Information Server web console	<ul style="list-style-type: none"> • Session • Persistent 	User name	<ul style="list-style-type: none"> • Session management • Authentication 	Cannot be disabled
InfoSphere Information Governance Catalog		<ul style="list-style-type: none"> • Session • Persistent 	<ul style="list-style-type: none"> • User name • Internal identifiers • State of the tree 	<ul style="list-style-type: none"> • Session management • Authentication • Single sign-on configuration 	Cannot be disabled
InfoSphere Information Analyzer	Data Rules stage in the InfoSphere DataStage and QualityStage Designer client	Session	Session ID	Session management	Cannot be disabled

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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.

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^{Link}, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACS^{Link} licensee of the United States Postal Service.

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

Index

A

account name
 in UniVerse stages 10
aggregating data 19

C

character set maps, defining
 UniData stages 6
co-resident data sources 3
command-line syntax
 conventions 27
commands
 syntax 27
connection parameters, setting
 UniData stages 6
Create table options dialog box 15
customer support
 contacting 31

D

Data Browser 7, 12
data sources, accessing
 UniData 5.2 4
 UniVerse
 co-resident 3
 local 2

E

environment variables 4

F

file names, UniData 6, 7
FROM clauses 18

G

generated queries
 UniVerse stages 14, 17
GROUP BY clauses 18, 19

H

HAVING clauses 18

K

key fields 17

L

legal notices 37
local data sources 2

M

meta data, importing
 from UniData 4
multiple data files (multilevel files) 23
multivalued/multi-subvalued data and
 associations 21

O

ODBC configuration file 1
ORDER BY clauses 18, 19

P

product accessibility
 accessibility 25
product documentation
 accessing 33

S

SELECT clauses 18
software services
 contacting 31
special characters
 in command-line syntax 27
SQL statements
 generated 14, 17
 syntax 17
 user-defined 14, 20
 viewing 12, 16
stages, editing
 UniData 5
 UniVerse 9
support
 customer 31
syntax
 command-line 27

T

table name
 in UniVerse stages 11, 16
trademarks
 list of 37
transaction control information 12

U

UniData 5.2 data sources
 handling large records 4
UniData data sources
 accessing 4
 importing meta data 4
UniData stages
 character set maps 6
 Columns tab 7
 connection parameters 6

UniData stages (*continued*)

 file name 6, 7
 input links 5, 6
 Inputs page 6
 NLS tab 6
 output links 5, 7
 Outputs page 7
 overview 5
 Selection tab 8
 Stage page 5
 update action 7
UniVerse data sources
 co-resident 3
 local 2
UniVerse stages
 account name 10
 Columns tab 11, 16
 connection parameters 10
 create table options 15
 Details tab 10
 Edit DDL tab 12
 Enter SQL tab 12, 15
 generated queries 14, 17
 input links 9, 11
 Inputs page 11
 NLS tab 9
 output links 9, 15
 Outputs page 16
 overview 9
 Selection tab 16, 18
 SQL Query tab 16
 Stage page 9
 table name 11, 16
 Transaction Handling tab 12, 16
 update action 11
 user-defined SQL statements 14, 20
 View SQL tab 12, 16, 17
update action
 in UniData stages 7
 in UniVerse stages 11
user-defined SQL statements 14, 20
UV/Net 2
UV.ACCOUNT file 2
uvodbc.config file 1

W

web sites
 non-IBM 29
WHERE clauses 18



Printed in USA

SC19-4260-00

