

# Ascential **DataStage**

ODBC Enterprise Stage User Guide  
Version 1.0

Part No. 00D-002ES10  
December 2004

This document, and the software described or referenced in it, are confidential and proprietary to Ascential Software Corporation ("Ascential"). They are provided under, and are subject to, the terms and conditions of a license agreement between Ascential and the licensee, and may not be transferred, disclosed, or otherwise provided to third parties, unless otherwise permitted by that agreement. No portion of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Ascential. The specifications and other information contained in this document for some purposes may not be complete, current, or correct, and are subject to change without notice. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT, INCLUDING WITHOUT LIMITATION STATEMENTS REGARDING CAPACITY, PERFORMANCE, OR SUITABILITY FOR USE OF PRODUCTS OR SOFTWARE DESCRIBED HEREIN, SHALL BE DEEMED TO BE A WARRANTY BY ASCENTIAL FOR ANY PURPOSE OR GIVE RISE TO ANY LIABILITY OF ASCENTIAL WHATSOEVER. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL ASCENTIAL BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs.

© 2004 Ascential Software Corporation. All rights reserved. DataStage®, EasyLogic®, EasyPath®, Enterprise Data Quality Management®, Iterations®, Matchware®, Mercator®, MetaBroker®, Application Integration, Simplified®, Ascential™, Ascential AuditStage™, Ascential DataStage™, Ascential ProfileStage™, Ascential QualityStage™, Ascential Enterprise Integration Suite™, Ascential Real-time Integration Services™, Ascential MetaStage™, and Ascential RTI™ are trademarks of Ascential Software Corporation or its affiliates and may be registered in the United States or other jurisdictions.

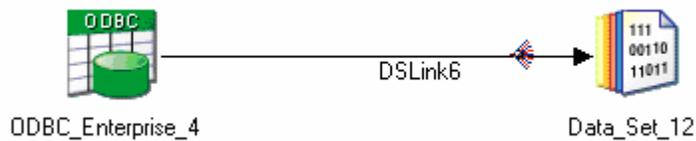
The software delivered to Licensee may contain third-party software code. See *Legal Notices (LegalNotices.pdf)* for more information.

# ODBC Enterprise Stage

The functionality of the ODBC Enterprise Stage for DataStage Enterprise Edition is to extend DataStage's capabilities to communicate with external data sources. Currently, connectivity to external data sources from DataStage's parallel canvas is limited to four databases: DB2, Informix, Oracle, and Teradata. ODBC Enterprise Stage has four modes. They are:

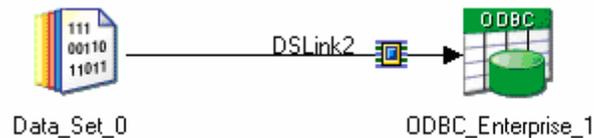
## Read

The ODBC Enterprise Stage in read mode reads records from an external data source table and places them in a DataStage data set.



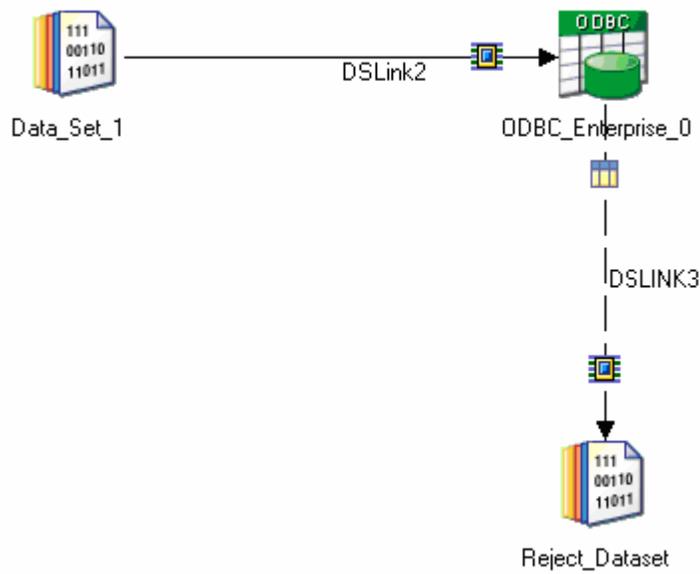
## Write

The ODBC Stage in write mode sets up a connection to external data source and inserts records into a table. The Stage takes a single input data set. The write mode determines how the records of a data set are inserted into the table.



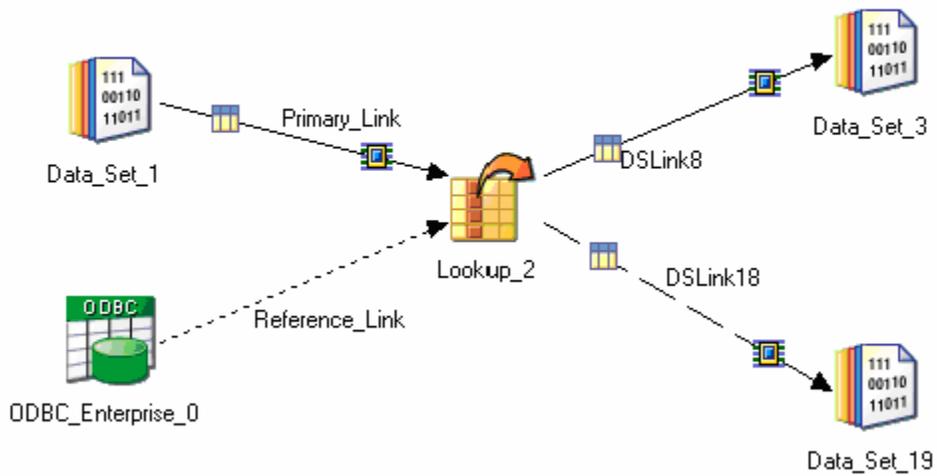
## Upsert

The ODBC Stage in upsert mode lets you insert into an External data source table or update an External data source table with data contained in a DataStage data set. You can match records based on field names and then update or insert those records.



## Lookup

The ODBC Stage in lookup mode lets you perform a join between an External data source table and a DataStage data set, with the resulting data output as a DataStage data set.



The stage editor has up to three pages, depending on whether you are accessing or transferring a file:

- **Stage Page.** This is always present and is used to specify general information about the stage.
- **Input Page.** This is present when you are accessing files from a Remote Host. This is where you specify details about the Input Link.
- **Output Page.** This is present when you are transferring data sets to a file. This is where you specify details about the Output Link.

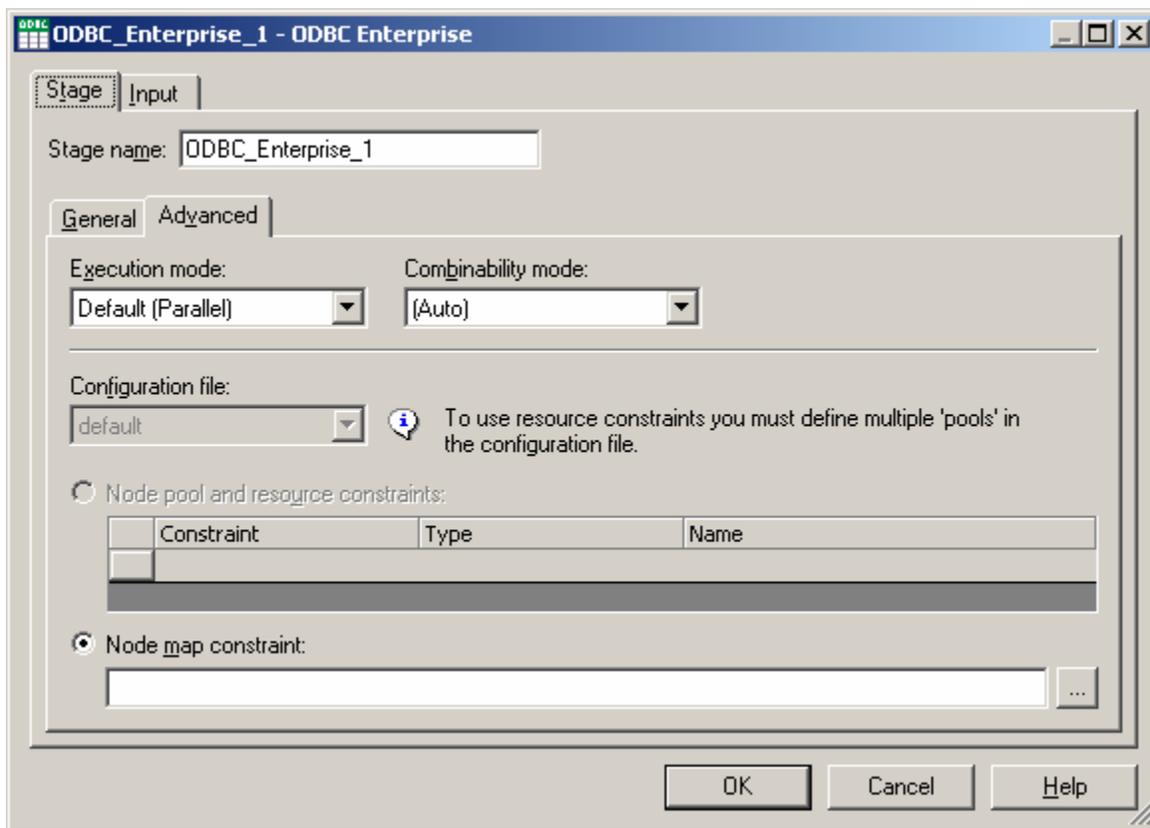
## Stage Page

### General Tab

The **General** tab allows you to specify an optional description of the stage.

### Advanced Tab

The **Advanced** tab allows you to set some particular features about how the stage behaves. Generally you can ignore this tab, and let the stage take the default values. It is intended for advanced users to finely tune operations. **Note** - The Execution mode is set automatically and cannot be changed. If the stage is only operating on one file (and there is one reader) the execution mode will be sequential. Otherwise it will be parallel.



## NLS Map Tab

The **NLS Map** (National Language Support) allows you to define a character set map for the ODBC stage. This overrides the default character set map for the project or the job. You can specify that the map be supplied as a job parameter if required. You can also select Allow per-column mapping. This allows character set maps to be specified for individual columns within the data processed by the ODBC stage. An extra property, NLS Map, appears in the Columns grid, but note that only ustring data types allow you to set an NLS map value (see Parallel Job Developer's Guide for details on data types).

## Input Page

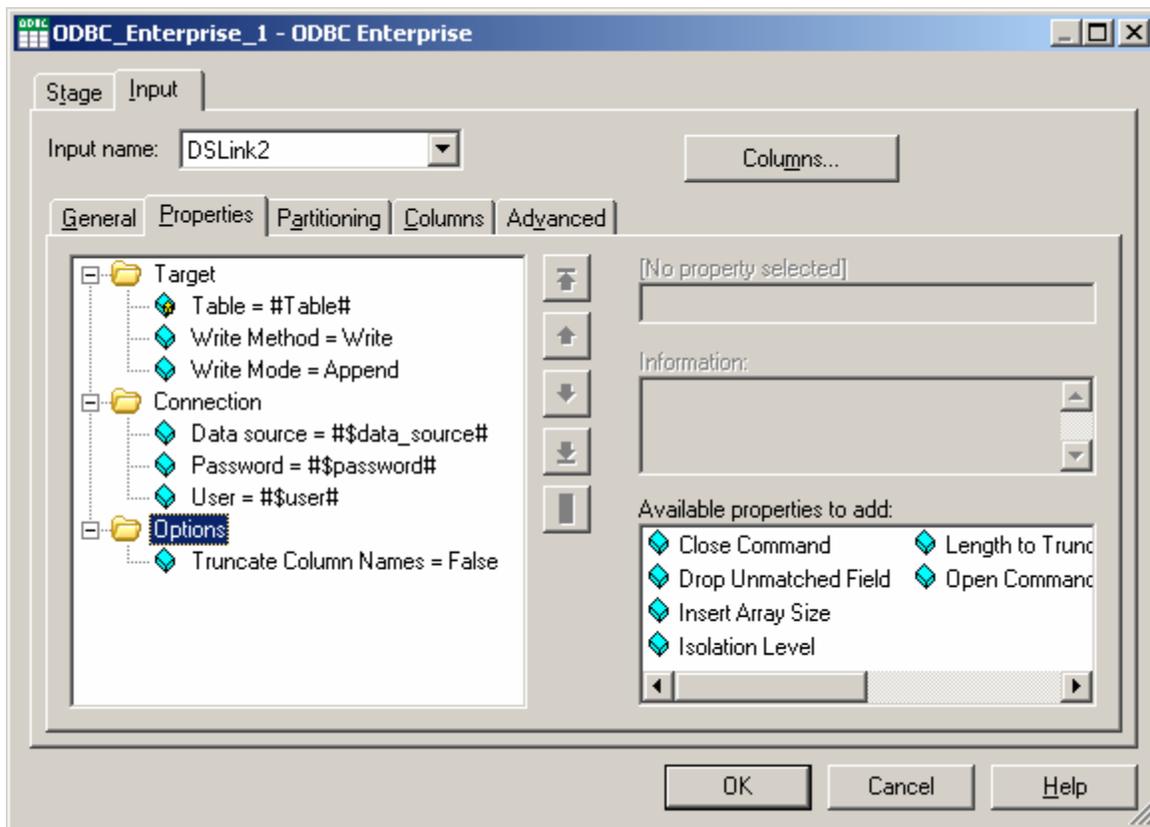
The input page is where you give information about the link going from a stage, which transfers files to a remote host using the ODBC Protocol. The Input page contains various tabs.

### Input Page in Write Mode

#### General Tab

The General tab allows you to enter an optional description of the link.

#### Properties Tab



The Properties tab allows you to specify properties, which determine what the stage actually does. The available properties are displayed in a tree structure. They are divided into categories to help you find your way around them. All the mandatory properties are included in the tree by default and cannot be removed. Properties that you must set a value for (i.e. which have not got a default value) are shown in a warning color (red by default), but changes to black when you have set a value.

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

Category/Property	Values	Default	Mandatory?	Repeats?	Dependent of
Target/ <a href="#">Table</a>	Name of the table	N/A	Y	N	Y
Target/ <a href="#">Write Method</a>	Upsert, Write	N/A	Y	N	N
Target/ <a href="#">Write Mode</a>	Append, Create, Replace, Truncate	N/A	Y	N	N
Connection/ <a href="#">Data source</a>	Data source	N/A	Y	N	Y
Connection/ <a href="#">User</a>	User name	N/A	Y	N	Y
Connection/ <a href="#">Password</a>	Password	N/A	Y	N	Y
Options/ <a href="#">Insert Array Size</a>	Integer	Y	N	N	N
Options/ <a href="#">Truncate Column Name</a>	False/True	Y	Y	N	N
Options/ <a href="#">Open Command</a>	Open Command	N	N	N	N
Options/ <a href="#">Close Command</a>	Close Command	N	N	N	N
Options/ <a href="#">Length to Truncate</a>	Integer	N	N	N	N
Options/ <a href="#">Isolation Level</a>	Read Uncommitted, Read Committed, Repeatable Read, Serializable	Y	N	N	N
Options/ <a href="#">Drop Unmatched Field</a>	True/False	Y	N	N	N

**Target Category –Table, Write Method and Write Mode** can be specified here.

**Table** connects the Stage to a Target file in a Remote host.

**Write Method** is used to write and export data into a single table; Must be set to Write to tell the stage to operate in write mode.

**Write Mode** The write mode of the Enterprise Stage determines how the records of the data set are inserted into the destination table. The write mode can have one of the following values:

- **Append:** This is the default mode. The table must exist and the record schema of the data set must be compatible with the table. The write Stage appends new rows to the table. The schema of the existing table determines the input interface of the Stage.
- **Create:** The Enterprise Stage creates a new table. If a table exists with the same name as the one being created, the step that contains the Enterprise Stage terminates with an error. The schema of the DataStage

data set determines the schema of the new table. The table is created with simple default properties. To create a table that is partitioned, indexed, in a non-default table space, or in some other non-standard way, you can use the -createstmt option with your own create table statement.

- **Replace:** The Enterprise Stage drops the existing table and creates a new one in its place. If a table exists with the same name as the one you want to create, it is overwritten. The schema of the DataStage data set determines the schema of the new table.
- **Truncate:** The Enterprise Stage retains the table attributes but discards existing records and appends new ones. The schema of the existing table determines the input interface of the Enterprise Stage. Each mode requires the specific user privileges.

**Connection** – the **Data source, Password** and **User** can be specified here.

Data source to be used for all database connections. This is mandatory. You can enter the name directly in its field or if you do not want the name to be hard coded, you can insert it from the Job Parameter Table and add an environment variable value at runtime for this job. You can also create a new variable. **Note** – the environment variables are taken from the operating system.

Enter the **password** in its field or if you do not want the password to be hard coded, you can insert it from the Job Parameter Table, and add an environment variable value at runtime for this job. You can also create a new variable. This is mandatory.

**Note** - Password is for login user. This is mandatory.

Enter the **User Name** directly in its field, or if you do not want the user name to be hard coded, you can insert it from the Job Parameter Table, This can be done in two ways.

- By entering values in the job parameter table, and
- By inserting the environment variable into the job parameter Table.

You can also create a new variable. **Note** – the environment variables are taken from the operating system.

**Options Category** – the **Insert Array Size, Truncate Column Name, Close Command, Length to Truncate, Drop unmatched field, Open Command and Isolation level** options can be specified here.

**Insert Array Size Type:** Enter the Integer value in this field, which specifies the Size of the insert host array. The default value is 2000 records.

**Truncate Column Name:** You can set one of two values, either **True** or **False**. Set True to truncate field names to the size allowed by the ODBC driver. Set False to disable the truncation of field names.

**Close Command:** Enter the SQL statement to be executed after insert array is processed. This statement is executed only once on the conductor node.

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

**Drop unmatched field:** You can set one of two values, either True or False. If this is set to True, unmatched fields of the dataset will be dropped. An unmatched field is a field for which there is no identically named field in the data source table. The default value is false.

**Open Command:** Enter the SQL statement to be executed before insert array is processed. This statement is executed only once on the conductor node.

**Isolation level:** Enter the Isolation level for accessing data. There are four Isolation levels; they are **Read Uncommitted**, **Read Committed**, **Repeatable Read** and **Serializable**. The default isolation level is decided by the database specified in the data source.

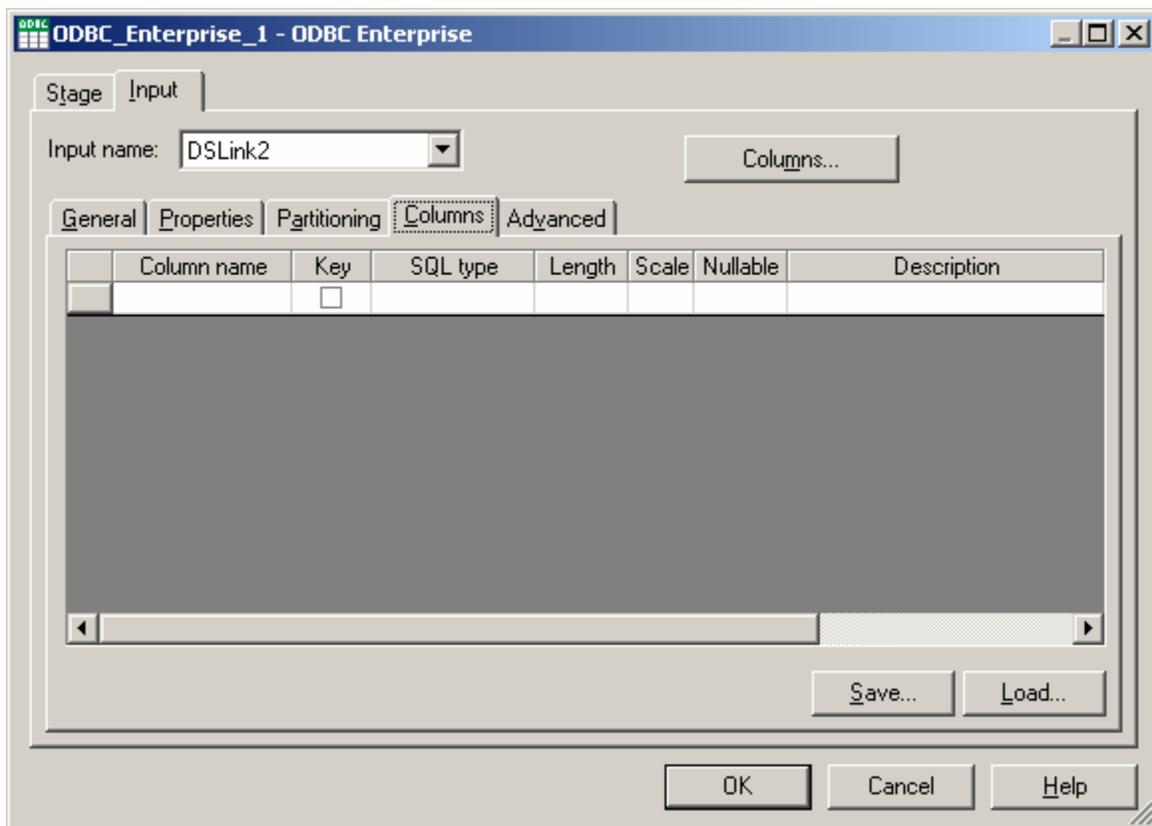
## Partitioning Tab

The Partitioning tab allows you to specify details about how the incoming data is partitioned or collected before it is operated on. It also allows you to specify that the data should be sorted before being operated on. By default, most stages partition in *Auto mode*. This attempts to work out the best partitioning method depending on execution modes of current and preceding stages and how many nodes are specified in the Configuration file. If the stage is operating in sequential mode, it will first collect the data before writing it to the file using the default Auto collection method.

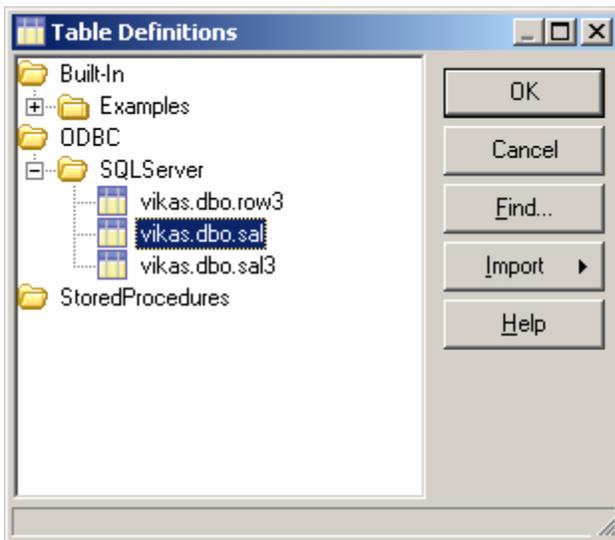
The Partitioning tab allows you to override this default behavior. The exact operation of this tab depends on:

1. Whether the stage is set to execute in parallel or sequential mode.
2. Whether the preceding stage in the job is set to execute in parallel or sequential mode.

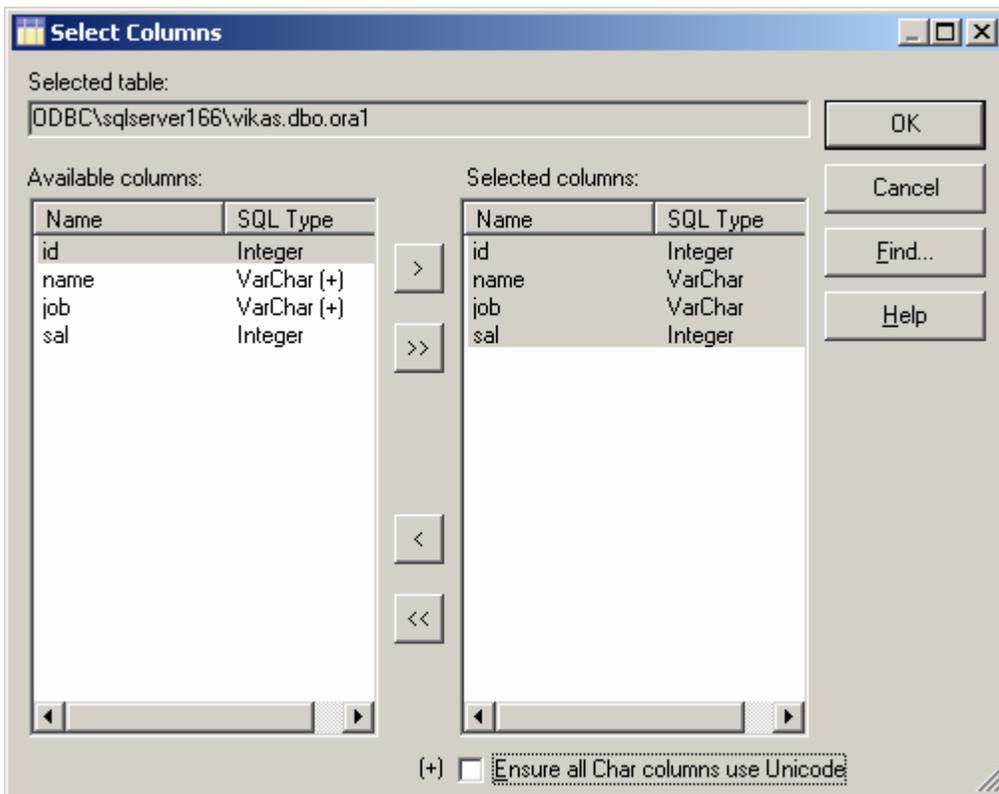
## Columns Tab



The **Input** and **Output** pages always have a **Columns** tab. This displays the column meta data for the selected link in a grid. The screen also contains **Save, Load, and Columns** buttons. Save button is used to save the values in the grid to the table. Load button loads the existing table using this popup wizard,



Click **Ok** after the appropriate table is selected and the following wizard is displayed.



Uncheck **Ensure all Char columns use Unicode** Check box

## **Advanced Tab**

The Advanced field allows you to specify how DataStage buffers data being input to or output from this stage. By default, DataStage buffers data in such a way that no deadlocks can arise; a deadlock being a situation where a number of stages are mutually dependent, and are waiting for input from another stage and cannot output until they have received it.

## **Columns Button**

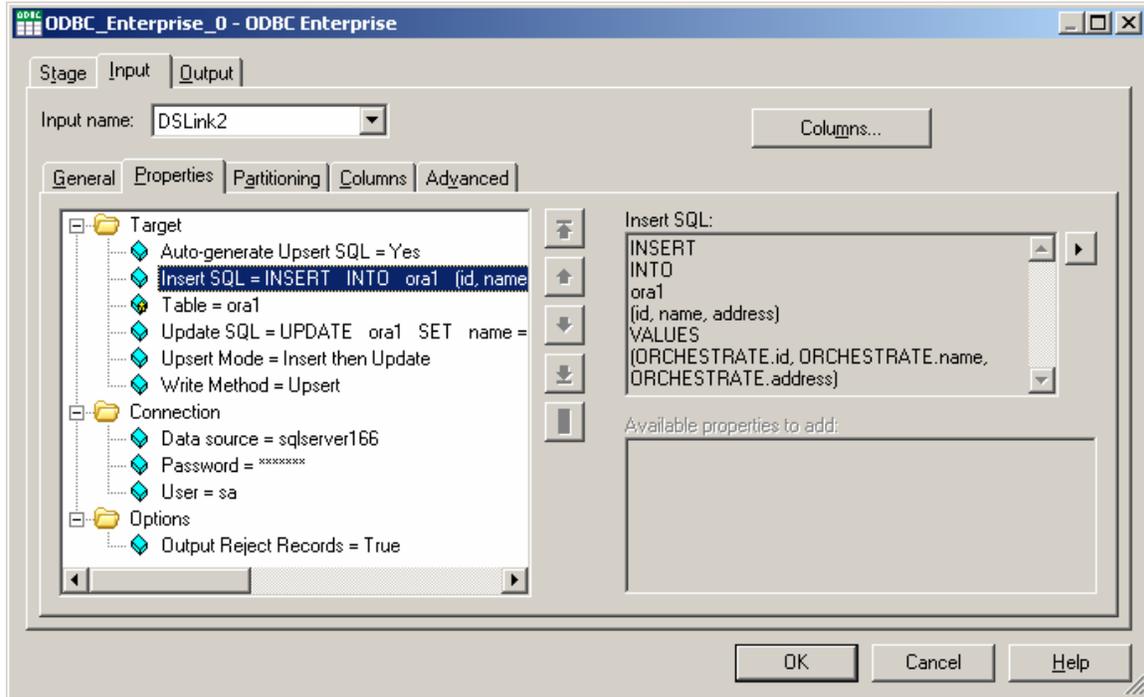
The columns button gives the list of column names.

## **Input Page in Upsert Mode**

### **General Tab**

The General tab allows you to enter an optional description of the link.

## Properties Tab



The Properties tab allows you to specify properties, which determine what the stage actually does. The available properties are displayed in a tree structure. They are divided into categories to help you find your way around them. All the mandatory properties are included in the tree by default and cannot be removed. Properties that you must set a value for (i.e. which have not got a default value) are shown in a warning color (red by default), but changes to black when you have set a value.

## Target Category

**Auto-generate Upsert SQL:** Specify whether the SQL is auto-generated from the set of loaded columns or not. When this option is set to true an additional option called table appears.

**Table:** Specify the name of the table to write to. You can specify a job parameter if required.

**Upsert mode:** This only appears for the Upsert write method. Allows you to specify how the insert and update statements are to be derived. Choose from:

1. If **Auto-generated Upsert SQL = YES**, the queries are generated from the built-in Templates. Choose from:
  - **Delete Only:** Select this to view an auto-generated Delete statement, DataStage generates a delete statement for you, based on the values supplied for tablename and columns details. The statement can be viewed by selecting the **Delete SQL** properties.

- **Delete then Insert:** DataStage generates delete and insert statements for you, based on the values supplied for table name and columns details. The statements can be viewed by selecting the **Delete SQL** or **Insert SQL** properties.
- **Insert Only:** Select this to view an auto-generated **Insert** statement, DataStage generates an insert statement for you, based on the values supplied for tablename and columns details. The statement can be viewed by selecting the **Insert SQL** properties.
- **Insert then Update:** DataStage generates insert and update statements for you, based on the values supplied for table name and columns details. The statements can be viewed by selecting the **Insert SQL** or **Update SQL** properties.
- **Update Only:** Select this to view an auto-generated **Update** statement, DataStage generates an update statement for you, based on the values supplied for tablename and columns details. The statement can be viewed by selecting the **Update SQL** properties.
- **Update then Insert:** DataStage generates update and insert statements for you, based on the values supplied for table name and columns details. The statements can be viewed by selecting the **Update SQL** or **Insert SQL** properties.

2. If **Auto-generated Upsert SQL** = No, the user has the privilege of defining the queries. Choose from:

- **Delete Only:** Select this to enter your own delete statement. Then select the **Delete SQL** property and edit the statement proforma.
- **Delete then Insert:** DataStage generates delete and insert statements for you, based on the values supplied for table name and columns details. The statements can be viewed by selecting the **Delete SQL** or **Insert SQL** properties.
- **Insert Only:** Select this to enter your own insert statement. Then select the **Insert SQL** property and edit the statement proforma.
- **Insert then Update:** DataStage generates insert and update statements for you, based on the values supplied for table name and columns details. The statements can be viewed by selecting the **Insert SQL** or **Update SQL** properties.
- **Update Only:** Select this to enter your own Update statement. Then select the **Update SQL** property and edit the statement proforma.
- **Update then Insert:** DataStage generates update and insert statements for you, based on the values supplied for table name and columns details. The statements can be viewed by selecting the **Update SQL** or **Insert SQL** properties.

**Write Method.** Must be set to upsert to choose upsert mode.

**Connection**– the **Data source, Password and user** options can be specified here.

**Data source** to be used for all database connections. This is mandatory. You can enter the name directly in its field or if you do not want the name to be hard coded, you can insert it from the Job Parameter Table and add an environment variable value at runtime for this job. You can also create a new variable. **Note** – the environment variables are taken from the operating system.

Enter the **password** in its field or if you do not want the password to be hard coded, you can insert it from the Job Parameter Table, and add an environment variable value at runtime for this job. You can also create a new variable. This is mandatory.

**Note** - Password is for login user. This is mandatory.

Enter the **User Name** directly in its field, or if you do not want the user name to be hard coded, you can insert it from the Job Parameter Table, This can be done in two ways.

- By entering values in the job parameter table, and
- By inserting the environment variable into the job parameter Table.

You can also create a new variable. **Note** – the environment variables are taken from the operating system.

## Options Category

### Open Command:

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

### Close Command:

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

### Output Reject Records:

This only appears for the Upsert write method. It is False by default, set to True to send rejected records to the reject link.

### Insert Array Size:

Enter the Integer value in this field, which specifies the Size of the insert host array. The default value is 2000 records.

## Partitioning Tab

The Partitioning tab allows you to specify details about how the incoming data is partitioned or collected before it is operated on. It also allows you to specify that the data should be sorted before being operated on. By default, most stages partition in *Auto mode*. This attempts to work out the best partitioning method depending on execution modes of current and preceding stages and how many nodes are specified in the Configuration file. If the stage is operating in sequential mode, it will first collect the data before writing it to the file using the default Auto collection method.

The Partitioning tab allows you to override this default behavior. The exact operation of this tab depends on:

- Whether the stage is set to execute in parallel or sequential mode.
- Whether the preceding stage in the job is set to execute in parallel or sequential mode.

## Columns Tab

The **Input** and **Output** pages always have a **Columns** tab. This displays the column meta data for the selected link in a grid.

## Advanced Tab

The Advanced field allows you to specify how DataStage buffers data being input to or output from this stage. By default, DataStage buffers data in such a way that no deadlocks can arise; a deadlock being a situation where a number of stages are mutually dependent, and are waiting for input from another stage and cannot output until they have received it.

## Columns Button

The columns button gives the list of column names.

# Output Page

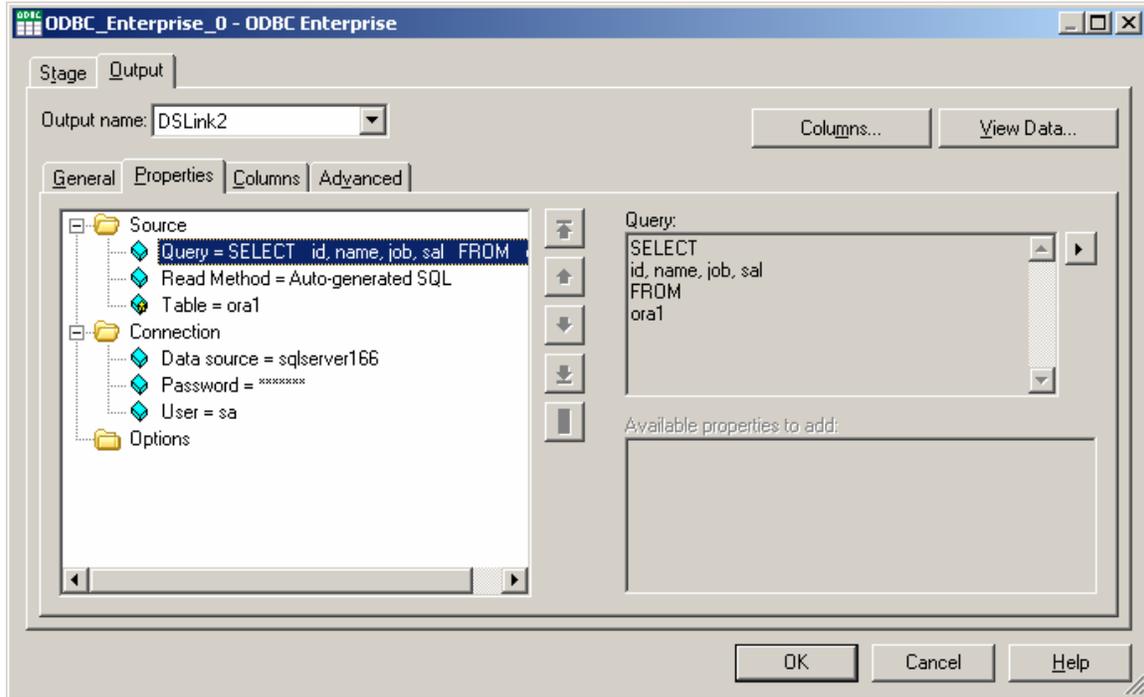
The output page is where you give information about the link from a remote host to a stage, which accesses data sets from files in the remote host using odbc. The output page contains various tabs. If your ODBC stage has a single output link, it is automatically in read mode. If your ODBC stage is connected to a Lookup stage by a reference link, it is automatically in lookup mode.

## Output Page in Read Mode

### General Tab

An optional description of the page can be given here.

## Properties Tab



The Properties tab in the Input and Output pages is the same except:

**Source option** – the **Read Method**, **Table** can be specified here.

### Read Method:

This property specifies whether you are specifying a table or a query when reading the ODBC database, and how you are generating the query.

- Select the **Table** method in order to use the **Table** property to specify the read.
- Select **Auto-generated SQL** to have DataStage automatically generate an SQL query based on the columns defined and the table you specify in the **Table** property.
- Select **User-defined SQL** to define your own query. A user-defined or auto-generated SQL will be read sequentially on one node.

**Query:** Optionally allows you to specify an SQL query to read a table. The query specifies the table and the processing that you want to perform on the table as it is read by the stage. This statement can contain joins, views, database links, synonyms, and so on.

**Table:** Specifies the name of the ODBC table. The table must exist and you must have SELECT privileges on the table. If your ODBC user name does not correspond to the owner of the specified table, you can prefix it with a table owner in the form:

table\_owner.table\_name

Table has dependent properties:

## Where

Specifies a WHERE clause of the SELECT statement to specify the rows of the table to include or exclude from the read operation. If you do not supply a WHERE clause, all rows are read.

## Select List

Optionally specifies an SQL select list, enclosed in single quotes that can be used to determine which columns are read. You must specify the columns in list in the same order as the columns are defined in the record schema of the input table.

**Connection**– the **Data source, Password and user** options can be specified here.

**Data source** to be used for all database connections. This is mandatory. You can enter the name directly in its field or if you do not want the name to be hard coded, you can insert it from the Job Parameter Table and add an environment variable value at runtime for this job. You can also create a new variable. **Note** – the environment variables are taken from the operating system.

Enter the **password** in its field or if you do not want the password to be hard coded, you can insert it from the Job Parameter Table, and add an environment variable value at runtime for this job. You can also create a new variable. This is mandatory.

**Note** - Password is for login user. This is mandatory.

Enter the **User Name** directly in its field, or if you do not want the user name to be hard coded, you can insert it from the Job Parameter Table, This can be done in two ways.

- By entering values in the job parameter table, and
- By inserting the environment variable into the job parameter Table.

You can also create a new variable. **Note** – the environment variables are taken from the operating system.

**Data source:** Data source to be used for all database connections.

**Password:** Encrypted Password for connecting to data source.

**User:** User name for connecting to data source.

**Options:** the **FetchArraySize, Isolation Level, Close Command and Open Commands** can be specified here.

**FetchArraySize:** Enter number of rows to retrieve during each fetch operation. Default is 1.

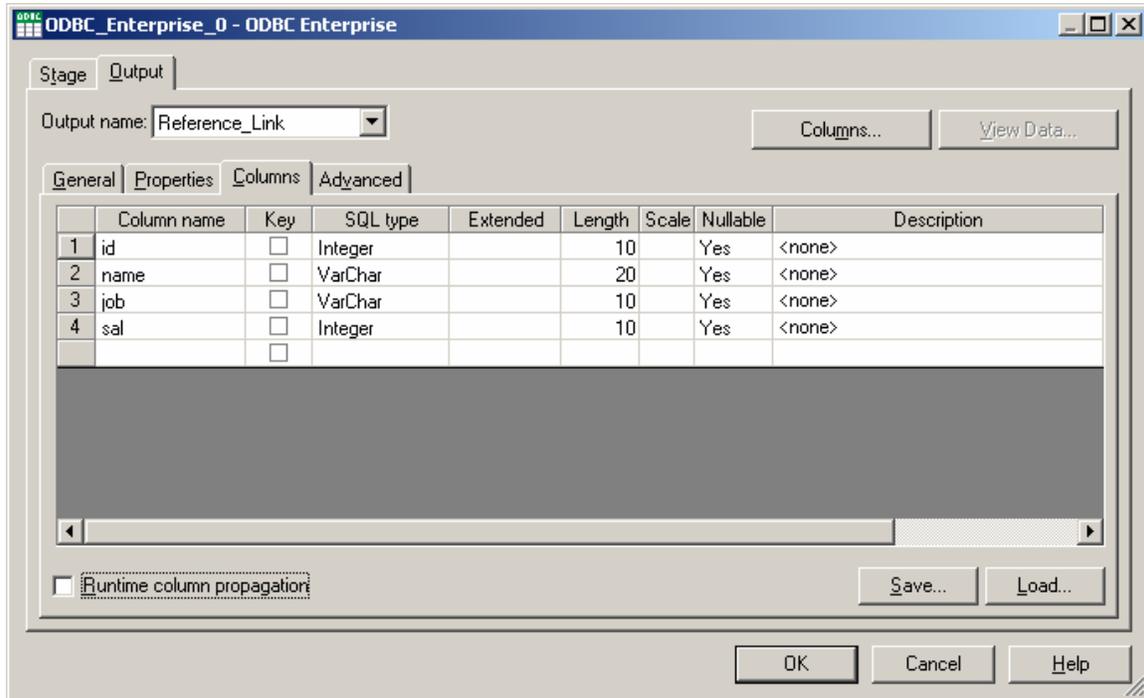
**Isolation Level:** Enter the Isolation Level for accessing the data. There are four Isolation levels; they are **Read Uncommitted, Read Committed, Repeatable Read** and **Serializable**. The default Isolation Level is decided by the database specified in the data source.

**Close Command:** Enter an SQL statement to be executed after the insert array is processed. You cannot commit work using this option. The statements are executed only once on the conductor node.

**Open Command:** Enter an SQL statement to be executed before the insert array is processed. The statements are executed only once on the conductor node.

## Columns Tab

The **Input** and **Output** pages always have a **Columns** tab. This displays the column meta data for the selected link in a grid.



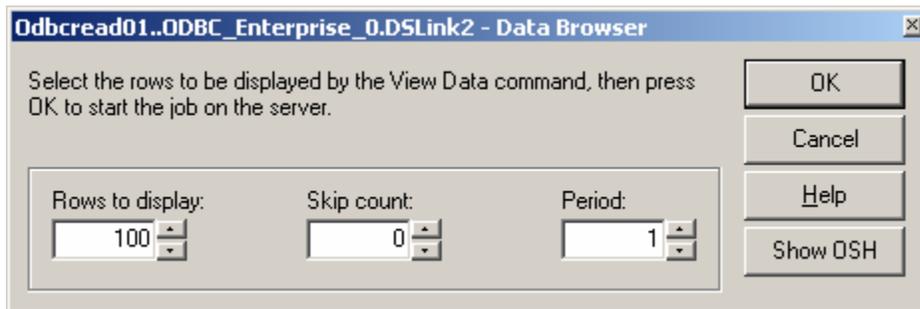
## Advanced Tab

The Advanced field allows you to specify how DataStage buffers data being input to or output from this stage. By default, DataStage buffers data in such a way that no deadlocks can arise; a deadlock being a situation where a number of stages are mutually dependent, and are waiting for input from another stage and cannot output until they have received it.

## Columns Button

The columns button gives the list of column names.

## View data Button



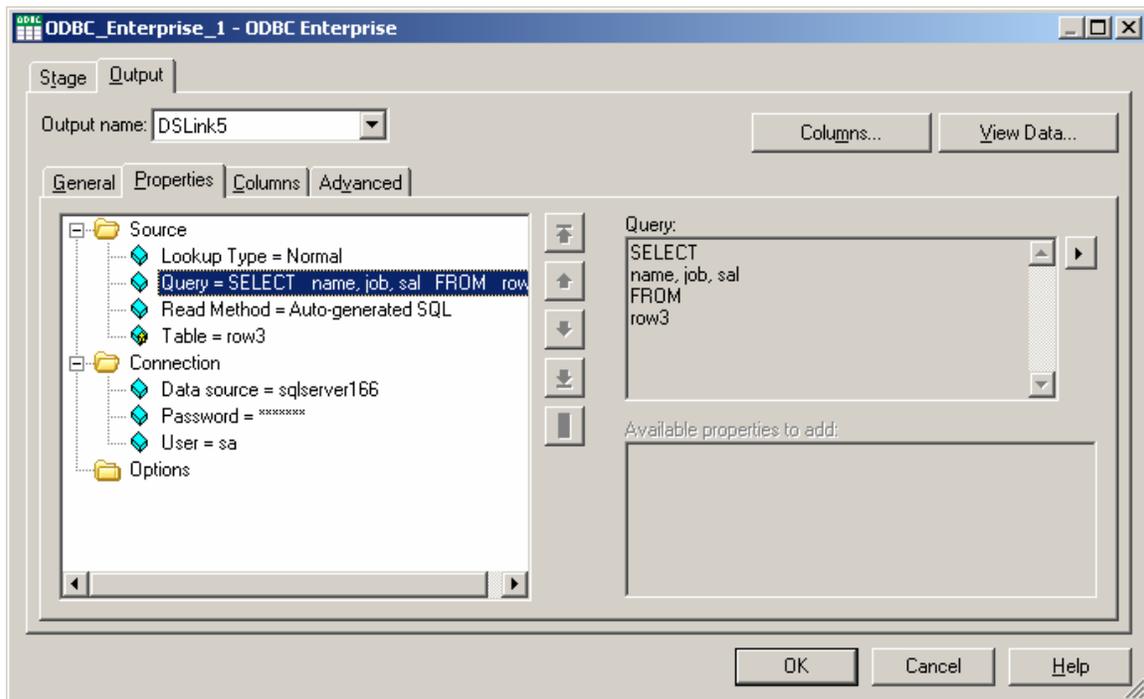
The View Data button can be used to view the URI file data.

## Output Page in Lookup Mode

### General Tab

An optional description of the page can be given here.

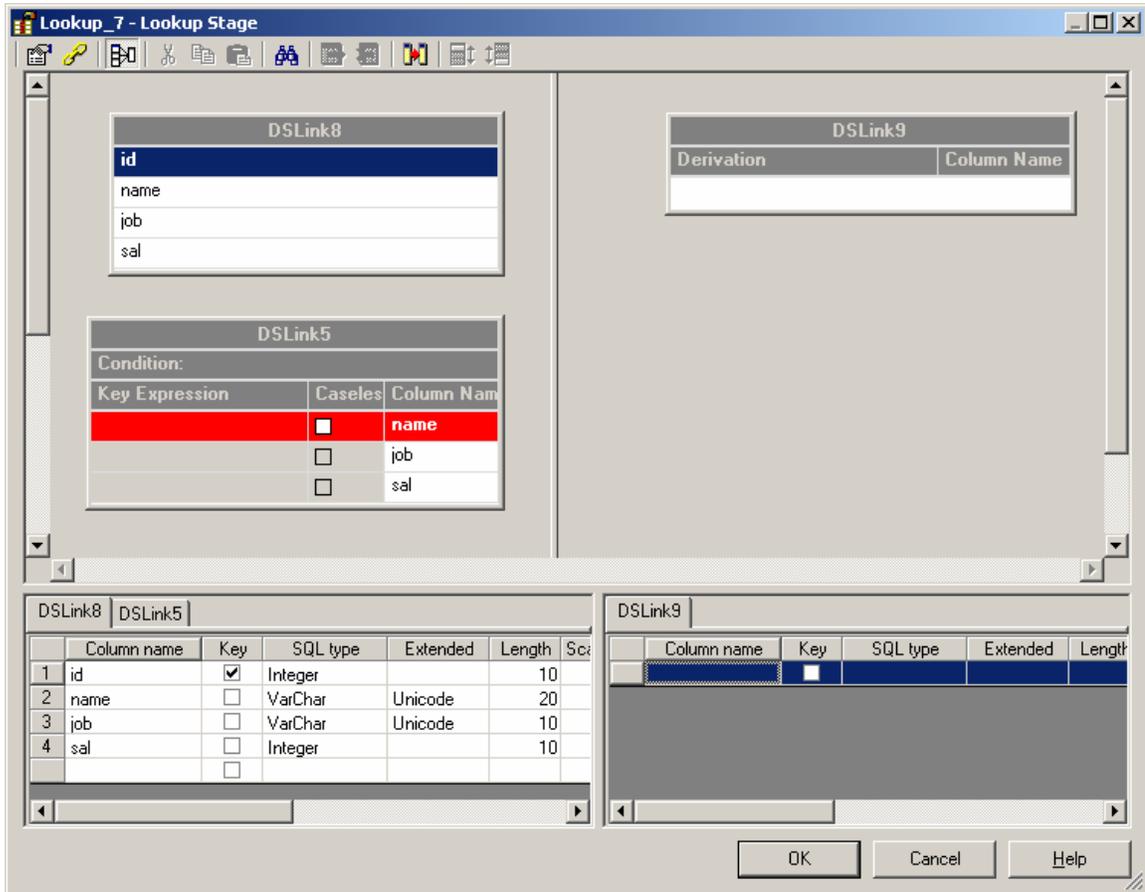
### Properties Tab



**Source option** contains **Table**, **Read Method** and an additional property **Lookup Type**

**Lookup Type.** For Normal lookups, ODBC Enterprise Stage will provide data for an in-memory look up (Lookup Type = Normal), for sparse lookups, the lookup will access the database directly (Lookup Type = Sparse). If the

Lookup Type is Normal, the Lookup stage can have multiple reference links. If the Lookup Type is Sparse, the Lookup stage can only have one reference link.



### Performing a Direct Lookup on an ODBC Database Table

Connect the ODBC Enterprise Stage to a Lookup stage using a reference link.

- In the **Output Link Properties Tab**:
  - Set the Lookup Type to Sparse.
  - Choose a Read Method. This is Table by default (which reads directly from a table), but you can also choose to read using auto-generated SQL or user-generated SQL.
  - Specify the table to be read for the lookup.
  - If using a Read Method of user-generated SQL, specify the SELECT SQL statement to use. DataStage provides the auto-generated statement, which can be edited as required.

Under the Connection category, you can either manually specify a connection string for server, user and password or if you do not want the values to be hard coded, you can insert it from the Job Parameter Table, or add an environment variable value at runtime for this job, you can also create a new variable.

- By default, DataStage connects to ODBC default database, but you can specify a database if required.
- Ensure column meta data has been specified for the write.

## Performing an In Memory Lookup on an ODBC Database Table

This is the default method. It has the same requirements as a direct lookup, except:

- In the **Output Link Properties Tab**:

Set the Lookup Type to Normal.

### Read Method.

This property specifies whether you are specifying a table or a query when reading the ODBC database, and how you are generating the query.

- Select the Table method in order to use the Table property to specify the read.
- Select **Auto-generated SQL** to have DataStage automatically generate an SQL query based on the columns defined and the table you specify in the Table property.
- Select **User-defined SQL** to define your own query. A user-defined or auto-generated SQL will be read sequentially on one node.

ODBC Enterprise Stage will provide data for an in-memory look up (Lookup Type = Normal) or whether the lookup will access the database directly (Lookup Type = Sparse). If the Lookup Type is Normal, the Lookup stage can have multiple reference links. If the Lookup Type is Sparse, the Lookup stage can only have one reference link.

**Query.** Optionally allows you to specify an SQL query to read a table. The query specifies the table and the processing that you want to perform on the table as it is read by the stage. This statement can contain joins, views, database links, synonyms, and so on.

**Table.** Specifies the name of the ODBC table. The table must exist and you must have SELECT privileges on the table. If your ODBC user name does not correspond to the owner of the specified table, you can prefix it with a table owner in the form:

table\_owner.table\_name

Table has dependent properties:

### Where

Specifies a WHERE clause of the SELECT statement to specify the rows of the table to include or exclude from the read operation. If you do not supply a WHERE clause, all rows are read.

### Select List

Optionally specifies an SQL select list, enclosed in single quotes that can be used to determine which columns are read. You must specify the columns in list in the same order as the columns are defined in the record schema of the input table.

**Connection**– the **Data source, Password and user** options can be specified here.

**Data source** to be used for all database connections. This is mandatory. You can enter the name directly in its field or if you do not want the name to be hard coded, you can insert it from the Job Parameter Table and add an environment variable value at runtime for this job. You can also create a new variable. **Note** – the environment variables are taken from the operating system.

Enter the **password** in its field or if you do not want the password to be hard coded, you can insert it from the Job Parameter Table, and add an environment variable value at runtime for this job You can also create a new variable. This is mandatory.

**Note** - Password is for login user. This is mandatory.

Enter the **User Name** directly in its field, or if you do not want the user name to be hard coded, you can insert it from the Job Parameter Table, This can be done in two ways.

- By entering values in the job parameter table, and
- By inserting the environment variable into the job parameter Table.

You can also create a new variable. **Note** – the environment variables are taken from the operating system.

**Options** the **FetchArraySize**, **Lookup Fail Action**, **Isolation Level**, **Close Command** and **Open Commands** can be specified here.

**FetchArraySize:** Enter number of rows to retrieve during each fetch operation. Default is 1.

**Isolation Level:** Enter the Isolation Level for accessing the data. There are four Isolation levels; they are **Read Uncommitted**, **Read Committed**, **Repeatable Read** and **Serializable** The default Isolation Level is decided by the database specified in the data source.

**Close Command:** Enter an SQL statement to be executed after the insert array is processed. You cannot commit work using this option. The statements are executed only once on the conductor node.

**Open Command:** Enter an SQL statement to be executed before the insert array is processed. The statements are executed only once on the conductor node.

## Columns Tab

The **Input** and **Output** pages always have a **Columns** tab. This displays the column meta data for the selected link in a grid.

## **Advanced Tab**

The Advanced field allows you to specify how DataStage buffers data being input to or output from this stage. By default, DataStage buffers data in such a way that no deadlocks can arise; a deadlock being a situation where a number of stages are mutually dependent, and are waiting for input from another stage and cannot output until they have received it.

## **Columns Button**

The columns button gives the list of column names.

## **Output Page for Reject Links**

If you have an output link as well as an input link on a stage in Read or Upsert mode, this is automatically a Rejects link.

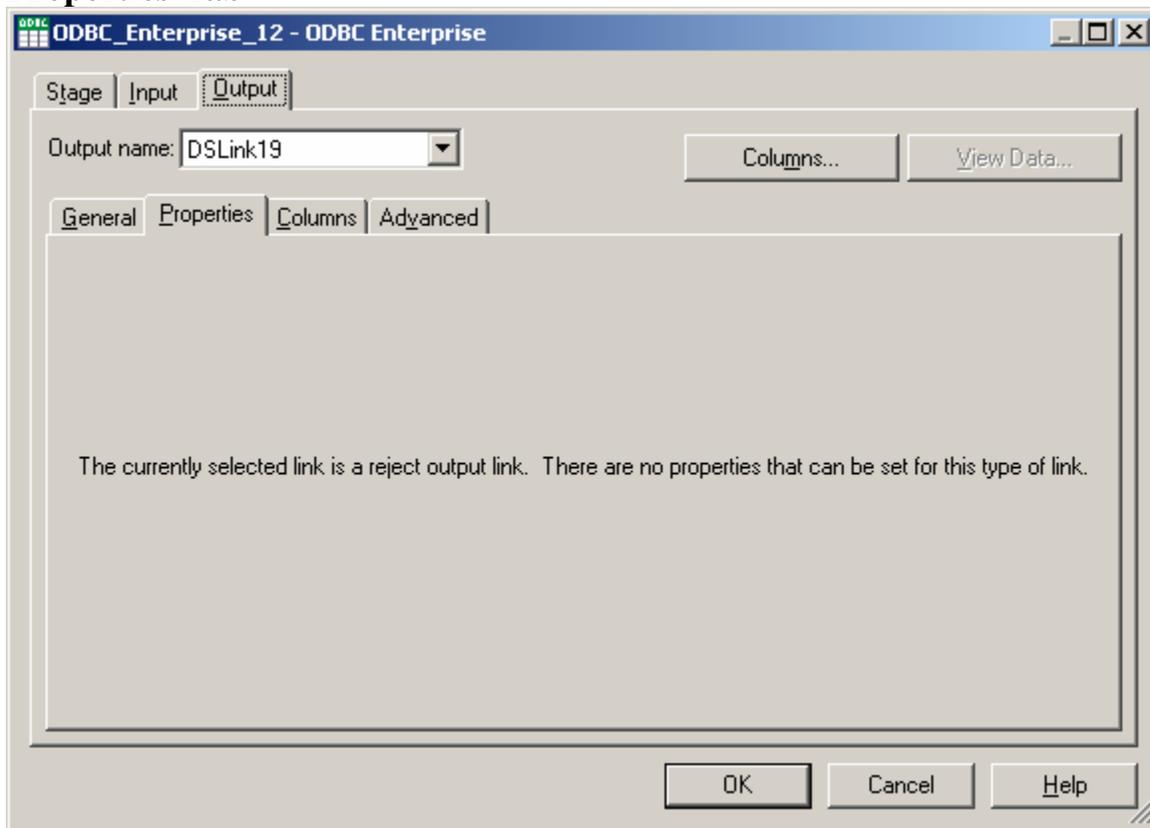
If you have more than one output link on a stage in Lookup or Write mode, you can choose which one is the reject link.

The pages and properties of the output page for a reject link are as follows.

## **General Tab**

The General tab allows you to enter an optional description of the link.

## Properties Tab



## Columns Tab

The **Input** and **Output** pages always have a **Columns** tab. This displays the column meta data for the selected link in a grid.

## Advanced Tab

The Advanced field allows you to specify how DataStage buffers data being input to or output from this stage. By default, DataStage buffers data in such a way that no deadlocks can arise; a deadlock being a situation where a number of stages are mutually dependent, and are waiting for input from another stage and cannot output until they have received it.

## Columns Button

The columns button gives the list of column names.