

Ascential DataStage™

Oracle OCI Stage Guide

Version 1.1



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 NONINFRINGEMENT 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, 2002-2003 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.

Adobe Acrobat is a trademark of Adobe Systems, Inc. Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Oracle, Oracle8i, Oracle9i, and Oracle 10g are either registered trademarks or trademarks of Oracle Corporation. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. Other marks mentioned are the property of the owners of those marks. The software delivered to Licensee may contain third-party software code. See *Legal Notices* ([LegalNotices.pdf](#)) for more information.

How to Use This Guide

The Oracle OCI stage reads from any Ascential DataStage stage into an Oracle database. It also writes data to any Ascential DataStage stage from an Oracle database. Version 1.1 of Oracle OCI is compatible with Ascential DataStage Release 7.5.1.

Audience

This guide is intended for DataStage designers who create or modify jobs that use the Oracle OCI stage.

How This Book is Organized

The following table lists topics that may be of interest to you and it provides links to these topics.

To learn about	Read...
Functionality	"Functionality" on page 2
Configuration requirements	"Configuration Requirements" on page 4
Installation	"Installing the Stage" on page 4
Defining the Oracle connection	"Defining the Oracle Connection" on page 5
Converting Oracle OCI 8 projects to Oracle OCI	"Converting an Oracle 8 Project to Oracle 9 or 10" on page 7
Defining input data	"Defining Input Data" on page 8
Writing data to Oracle	"Writing Data to Oracle" on page 21
Defining output data	"Defining Output Data" on page 23
Reading data from Oracle	"Reading Data from Oracle" on page 31
DATE data type considerations	"DATE Data Type Considerations" on page 33

To learn about	Read...
Oracle data type support	"Oracle Data Type Support" on page 34
Handling \$ and # characters	"Handling \$ and # Characters" on page 36

Related Documentation

To learn more about documentation from other Ascential products and third-party documentation as they relate to Oracle OCI, refer to the following sections/tables.

Ascential Software Documentation

Guide	Description
<i>Ascential DataStage Designer Guide</i>	General principles for designing jobs
<i>Ascential DataStage Server Job Developer's Guide</i>	Techniques for designing server jobs
<i>Ascential DataStage Manager Guide</i>	Techniques for using and maintaining the DataStage Repository
<i>Ascential MetaStage User's Guide</i>	Information about Ascential MetaStage™
<i>Ascential DataStage NLS Guide</i>	Information about NLS and techniques for character-set mapping
<i>Ascential DataStage Plug-In Installation and Configuration Guide</i>	Information required to configure your system and install this stage

Third-Party Documentation

Guide	Description
<i>Programmer's Guide to the Oracle Call Interface</i>	Information about Oracle including information about installation of Oracle

Conventions

Convention	Used for...
bold	Field names, button names, menu items, and keystrokes. Also used to indicate filenames, and window and dialog box names.
<code>user input</code>	Information that you need to enter as is.
<code>code</code>	Code examples
<i>variable</i> or <variable>	Placeholders for information that you need to enter. Do not type the greater-/less-than brackets as part of the variable.
>	Indicators used to separate menu options, such as: Start >Programs >Ascential DataStage
[A]	Options in command syntax. Do not type the brackets as part of the option.
B...	Elements that can repeat.
A B	Indicator used to separate mutually-exclusive elements.
{ }	Indicator used to identify sets of choices.

Contacting Support

To reach Customer Care, please refer to the information below:

Call toll-free: 1-866-INFONOW (1-866-463-6669)

Email: support@ascentialsoftware.com

Ascential Developer Net: <http://developernet.ascential.com>

Please consult your support agreement for the location and availability of customer support personnel.

To find the location and telephone number of the nearest Ascential Software office outside of North America, please visit the Ascential Software Corporation website at <http://www.ascentialsoftware.com>.

Contents

Audience	iii
How This Book is Organized	iii
Related Documentation	iv
Ascential Software Documentation	iv
Third-Party Documentation	iv
Conventions	v
Contacting Support	v
Introduction	1
Functionality	2
Configuration Requirements	4
Installing the Stage	4
Defining the Oracle Connection	5
Connecting to an Oracle Database	6
Defining Character Set Mapping	7
Converting an Oracle 8 Project to Oracle 9 or 10	7
Defining Input Data	8
About the Input Page	9
Reject Row Handling	21
Writing Data to Oracle	21
Using Generated SQL Statements	22
Using User-Defined SQL Statements	23
Defining Output Data	23
About the Output Page	24

Reading Data from Oracle	31
Using the SQL Builder	31
Using Generated Queries	31
Using User-Defined Queries	33
DATE Data Type Considerations	33
Oracle Data Type Support	34
Character Data Types	34
Numeric Data Types	35
Date Data Types	35
Miscellaneous Data Types	36
Handling \$ and # Characters	36

Introduction

Version 1.1 of Oracle OCI adapts to a great variety of customer environments.

Version 1.1 of Oracle OCI results in a greater dependency on the *libclntsh* shared library, which is created during the installation of the Oracle client software. You must include the location containing this shared library in the shared library search path for Ascential DataStage jobs to run successfully using the new stage.

Oracle offers a proprietary call interface for C and C++ programmers that allows manipulation of data in an Oracle database. Oracle Call Interface (OCI) can connect and process SQL statements in the native Oracle environment without needing an external driver or driver manager. To use the Oracle OCI stage, you need only to install the Oracle Version 9.*n* or 10*g* client, which uses SQL*Net to access the Oracle server.

Oracle OCI works with Oracle Version 9*i* and 10*g* servers, provided you install the appropriate Oracle 9*i* or 10*g* software. For information about exceptions to this, see *Programmer's Guide to the Oracle Call Interface* for the appropriate release.

With Oracle OCI, you can:

- Generate your SQL statement.
- Use a file name to contain your SQL statement.
- Clear a table before loading using a TRUNCATE statement. (Clear table)
- Choose how often to commit rows to the database. (Transaction size)
- Input multiple rows of data in one call to the database. (Array size)
- Read multiple rows of data in one call from the database. (Array size)
- Specify transaction isolation levels for concurrency control and transaction performance tuning. (Transaction Isolation)
- Specify criteria that data must meet before being selected. (WHERE clause)
- Specify criteria to sort, summarize, and aggregate data. (Other clauses)
- Specify the behavior of parameter `marks` in SQL statements. For information on the Pre 4.2 user-defined SQL behavior, see ["Defining Input Data"](#) on page 8 and ["Defining Output Data"](#) on page 23

Oracle OCI lets you rapidly and efficiently prepare and load streams of tabular data from any Ascential DataStage stage (for example, the ODBC stage, the Sequential File stage, and so forth) to and from tables of the target Oracle database. The Oracle client on Windows or UNIX uses SQL*Net to access an Oracle server on Windows or UNIX.

Each Oracle OCI stage is a passive stage that can have any number of input, output, and reference output links:

- Input links specify the data you are writing, which is a stream of rows to be loaded into an Oracle database. You can specify the data on an input link using an SQL statement constructed by Ascential DataStage or a user-defined SQL statement.
- Output links specify the data you are extracting, which is a stream of rows to be read from an Oracle database. You can also specify the data on an output link using an SQL statement constructed by Ascential DataStage or a user-defined SQL statement.
- Each reference output link represents a row that is key read from an Oracle database (that is, it reads the record using the key field in the WHERE clause of the SQL SELECT statement).

For more information on using a stage in a job, see *Ascential DataStage Server Job Developer's Guide*.

Functionality

Oracle OCI has the following functionality and benefits:

- Support for transaction grouping to control a group of input links from a Transformer stage. This lets you write a set of data to a database in one transaction. Oracle OCI opens one database session per transaction group.
- Support for reject row handling. Link reject variables tell the Transformer stage the Oracle DBMS error code when an error occurs in the Oracle OCI stage for insert, update, and so forth, for control of job execution. The format of the error is DBMS.CODE=ORA-xxxxx. For more information on how the Transformer stage works with link reject variables, see *Ascential DataStage Server Job Developer's Guide*.
- Support for create and drop table functionality before writing to a table.
- Support for before and after SQL statements to execute user-defined SQL statements before or after the stage writes or reads into a database.

- Support of stream input, stream output, and reference output links.
- The ability to use the **Derivation** cell to specify fully-qualified column names used to construct an SQL SELECT statement for output and reference links.
Note When you select **Enable case sensitive table/column name**, it is your responsibility to use quotation marks for the owner/ table.column name in the **Derivation** cell to preserve any lower-case letters.
- Performance and scalability benefits by using Oracle OCI rather than the ODBC stage to access Oracle tables.
- Prefetching of SELECT statement result set rows when executing a query. This minimizes server round trips and enhances performance.
- Reduction of the number of network round trips (more processing is done on the client).
- Support of new transparent data structures and interfaces.
- Elimination of open and close cursor round trips.
- Improved error handling.
- Use of Oracle OCI as a supplement to existing jobs that already use the ODBC stage, rather than as a replacement for the ODBC stage.
- Importing of table definitions. For more information about meta data import, see *Ascential DataStage Server Job Developer's Guide*.
- Support of a file name to contain your SQL statement.
- Support for NLS (National Language Support). For more information, see *Ascential DataStage Administrator Guide* or *NLS Guide*.
- Support for Ascential MetaStage™. For information, see *Ascential MetaStage User's Guide*.
- Support for foreign key meta data import. For information, see *Ascential DataStage Manager Guide* and *Ascential DataStage Designer Guide*.
- Support for the behavior of parameter marks for SQL statements, which is the same as that for releases of Ascential DataStage before 4.2.

The following functionality is not supported:

- Co-existence of Oracle OCI 8i and Oracle OCI in the same job.
- Bulk loading for stream input links. Use the Oracle OCI Load stage to bulk load data into Oracle databases. For more information respectively, see *Ascential DataStage Server Job Developer's Guide* or the technical bulletin *Ascential DataStage Oracle OCI Load (74-0141)*.
- Stored procedures.
- Support of Oracle data types such as BLOB, FILE, LOB, LONG, LONG RAW, MSLABEL, OBJECT, RAW, REF, ROWID, or a named data type.

Configuration Requirements

Oracle OCI requires the following configuration:

- Version 9.*n* or 10*g* of the Oracle client software on the Ascential DataStage server machine, which requires one of the following:
 - **Version 9*i*:** Oracle9*i* Client (runtime)
 - **Version 10*g*:** Oracle 10*g* Client (runtime)
- Note** AIX 5.1 requires version 9.2 or later of the Oracle client software.
- Configuration of SQL*Net using a configuration program, for example, SQL*Net Easy Configuration, to set up and add database aliases.

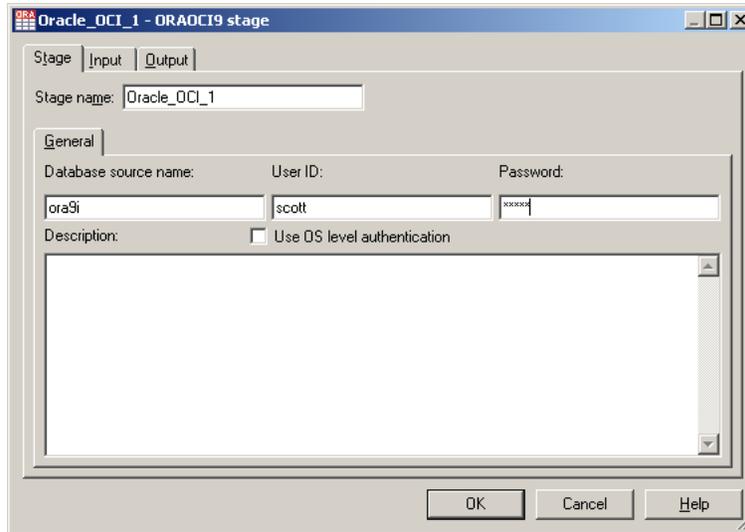
Installing the Stage

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

Note Ascential Software performed all testing of Oracle OCI using Windows and UNIX Oracle clients and Oracle servers only.

Defining the Oracle Connection

When you use the client GUI to edit an Oracle OCI stage, the **ORAOCI9Stage** dialog box appears:



This dialog box can have up to three pages (depending on whether there are inputs to and outputs from the stage):

- **Stage.** This page displays the name of the stage you are editing. The **General** tab defines the Oracle database source and logon information to connect to an Oracle database. For details see the following section, ["Connecting to an Oracle Database"](#) on page 6

The **NLS** tab defines a character set map to be used with the stage. (The **NLS** tab appears only if you have installed NLS.) For details, see ["Defining Character Set Mapping"](#) on page 7
- **Input.** This page is displayed only if you have an input link to this stage. It specifies the SQL table to use and the associated column definitions for each data input link. This page also specifies the type of update action and transaction isolation level information for concurrency control and performance tuning. It also contains the SQL statement used to write the data and lets you enable case sensitivity for SQL statements.
- **Output.** This page is displayed only if you have an output link to this stage. It specifies the SQL tables to use and the associated column definitions for each data output link. This page also specifies the type of query and transaction isolation level information for concurrency control and performance tuning. It also contains the SQL SELECT statement used to extract the data, and lets you enable case sensitivity for SQL statements.

To edit an Oracle OCI stage from the **ORAOCI9 Stage** dialog box:

- 1 Define the connection (see the following section).
- 2 Optional. [Define a character set map](#). See [page 7](#).
- 3 [Define the data on the input links](#). See [page 8](#).
- 4 [Define the data on the output links](#). See [page 23](#).

Connecting to an Oracle Database

Set the Oracle connection parameters on the **General** tab on the **Stage** page of the GUI. To connect to an Oracle database:

- 1 Enter the name of the Oracle database alias to access in the **Database source name** field. (This is the name you created using the Oracle Configuration Assistant.) Unless the database has a guest account, **User ID** must be a valid user in the database, have an alias in the database, or be a system administrator or system security officer. There is no default.
- 2 Enter the user name to use to connect to the Oracle database in the **User ID** field. This user must have sufficient privileges to access the specified database and source and target tables. This field is required except when **Use OS level authentication** is selected. There is no default.
- 3 Enter the password that is associated with the specified user name to use in the **Password** field. This field is required except when **Use OS level authentication** is selected. There is no default.
- 4 Choose an appropriate transaction isolation level to use from the **Transaction Isolation** list on the **General** tab on the **Input** page or **Output** page (see the **General** tab in "[Defining Input Data](#)" on [page 8](#) or "[Defining Output Data](#)" on [page 23](#)). This level provides the necessary consistency and concurrency control between transactions in the job and other transactions for optimal performance. Because Oracle does not prevent other transactions from modifying the data read by a query, that data may be changed by other transactions between two executions of the query. Thus, a transaction that executes a given query twice may experience both nonrepeatable reads and phantoms. Use one of the following transaction isolation levels:

Read Committed. Takes exclusive locks on modified data and sharable locks on all other data. Read committed is the default ISO level for all transactions.

Serializable. Takes exclusive locks on modified data and sharable locks on all other data. Serializable transactions see only those changes that were committed at the time the transaction began.

For more information on using these levels, see your Oracle documentation.

- 5 Enter an optional description of the Oracle OCI stage in the **Description** field.
- 6 Select **Use OS level authentication** to automatically log on using your operating system user name and password. It is cleared by default. For further details on Oracle login information, see your Oracle documentation.

Defining Character Set Mapping

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

Specify information using the following fields:

- **Map name to use with stage.** Defines the default character set map for the project or the job. You can change the map by selecting a map name from the list.
- **Show all maps.** Lists all the maps that are shipped with Ascential DataStage.
- **Loaded maps only.** Lists only the maps that are currently loaded.
- **Use Job Parameter....** Specifies parameter values for the job. Use the format *#Param#*, where *Param* is the name of the job parameter. The string *#Param#* is replaced by the job parameter when the job is run.

For more information about NLS or job parameters, see *Ascential DataStage NLS Guide* or *Ascential DataStage Designer Guide*.

Converting an Oracle 8 Project to Oracle 9 or 10

The OCI8TO9.B utility allows you to migrate any Oracle OCI 8 stage in your jobs to an Oracle OCI stage. This utility can be run on Ascential DataStage release 6.0 and later. The utility is located in the utilities/unsupported directory. Before beginning, create a backup of the project to be converted. Also ensure no other user is accessing the project at the time of conversion.

Use ftp in binary mode to move the utility to your Ascential DataStage server, placing it into a temporary directory. Copy the utility into the

DSU_BP.O directory of each project that you want to convert. In addition, on Unix you need to run

```
chmod 750 OCI8TO9.B
```

Start Ascential DataStage Administrator from any DataStage Client PC. Select the **Project** tab, highlight the project you are converting, and click **Command**. At the **Command** prompt, type

```
RUN DSU_BP OCI8TO9.B
```

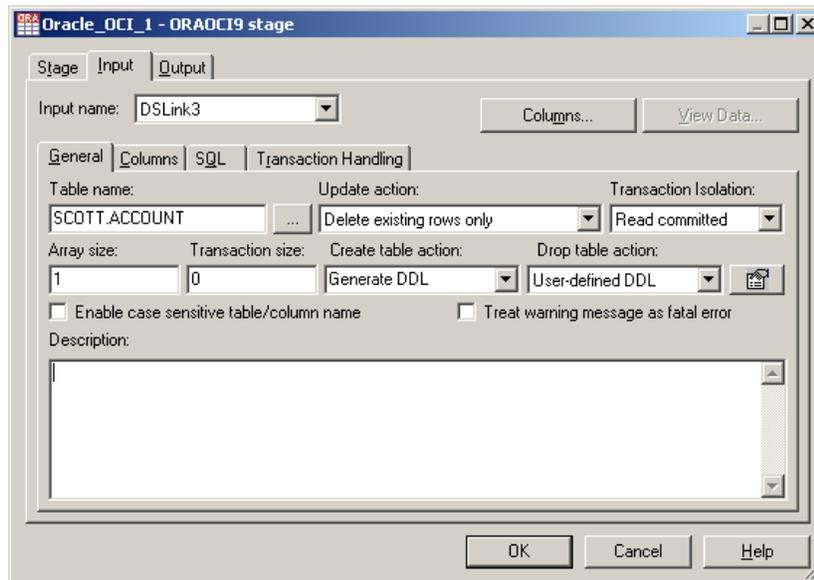
Click **Execute**. You are prompted to continue or exit. Reply with either 'Y' or 'N' and click **Respond**. If continuing, the output will pause at end-of-page by default; you can either click **Next** after each pause or clear **Pause....**

The utility creates a report of the jobs that were converted in the &COMO& directory called OCI8TO9.CONV.

Defining Input Data

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

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



About the Input Page

The **Input** page has an **Input name** field; the **General**, **Columns, SQL**, and **Transaction Handling** tabs; and the **Columns...** and **View Data...** buttons:

- **Input name.** The name of the input link. Choose the link you want to edit from the **Input name** list. This list displays all the input links to the Oracle OCI stage.
- Click **Columns...** to display a brief list of the columns designated on the input link. As you enter detailed meta data in the **Columns** tab, you can leave this list displayed.
- Click **View Data...** to invoke the Data Browser. This lets you look at the data associated with the input link in the database. For a description of the Data Browser, see *Ascential DataStage Designer Guide*.

General Tab

This tab is displayed by default. It contains the following fields:

- **Table name.** This required field is editable when the update action is *not* **User-defined SQL** (otherwise, it is read-only). It is the name of the target Oracle table the data is written to, and the table must exist or be created by choosing **generate DDL** from the **Create table action** list. You must have insert, update, or delete privileges, depending on input mode. You must specify **Table name** if you do not specify **User-defined SQL**. There is no default.

Click ... (Browse button) to browse the Repository to select the table.

- **Update action.** Specifies which SQL statements are used to update the target table. Some update actions require key columns to update or delete rows. There is no default. Choose the option you want from the list:
 - **Clear table then insert rows.** Deletes the contents of the table and adds the new rows, with slower performance because of transaction logging.
 - **Truncate table then insert rows.** Truncates the table with no transaction logging and faster performance.
 - **Insert rows without clearing.** Inserts the new rows in the table.
 - **Delete existing rows only.** Deletes existing rows in the target table that have identical keys in the source files.

- **Replace existing rows completely.** Deletes the existing rows, then adds the new rows to the table.
- **Update existing rows only.** Updates the existing data rows. Any rows in the data that do not exist in the table are ignored.
- **Update existing rows or insert new rows.** Updates the existing data rows before adding new rows. It is faster to update first when you have a large number of records.
- **Insert new rows or update existing rows.** Inserts the new rows before updating existing rows. It is faster to insert first if you have only a few records.
- **User-defined SQL.** Writes the data using a user-defined SQL statement, which overrides the default SQL statement generated by the stage. If you choose this option, you enter the SQL statement on the SQL tab. See "[Using User-Defined SQL Statements](#)" on page 23 for details.
- **User-defined SQL file.** Reads the contents of the specified file to write the data.
- **Transaction Isolation.** Provides the necessary concurrency control between transactions in the job and other transactions. Use one of the following transaction isolation levels:
 - **Read committed.** Takes exclusive locks on modified data and sharable locks on all other data. Each query executed by a transaction sees only data that was committed before the query (not the transaction) began. Oracle queries never read dirty (uncommitted) data. This is the default.
 - **Serializable.** Takes exclusive locks on modified data and sharable locks on all other data. Serializable transactions see only the changes that were committed at the time the transaction began.
- Note** If **Enable transaction grouping** is selected on the [Transaction Handling tab](#) (see [page 18](#)), only the **Transaction Isolation** value for the first link is used for the entire group.
- **Array size.** Specifies the number of rows to be transferred in one call between Ascential DataStage and Oracle before they are written. Enter a positive integer to indicate how often Oracle performs writes to the database. The default value is 1, that is, each row is written in a separate statement.

Larger numbers use more memory on the client to cache the rows. This minimizes server round trips and maximizes performance by executing fewer statements. If this number is too large, the client may run out of memory.

Array size has implications for Ascential DataStage's handling of reject rows. See "[Reject Row Handling](#)" on page 21.

- **Transaction size.** This field exists for backward compatibility, but it is ignored for version 3.0 and later of the Plug-in. The transaction size for new jobs is now handled by **Rows per transaction** on the **Transaction Handling** tab.
- **Create table action.** Creates the target table in the specified database if **Generate DDL** is selected. It uses the column definitions in the **Columns** tab and the table name and the TABLESPACE and STORAGE properties for the target table. The generated Create Table statement includes the TABLESPACE and STORAGE keywords, which indicate the location where the table is created and the storage expression for the Oracle storage-clause. You must have CREATE TABLE privileges on your schema.

You can also specify your own CREATE TABLE SQL statement. You must enter the storage clause in Oracle format. (Use the **User-defined DDL** tab on the **SQL** tab for a complex statement.)

Choose one of the following options to create the table:

- **Do not create target table.** Specifies that the target table is not created, and the Drop table action field and the Create Table Properties button on the right of the dialog are disabled.
- **Generate DDL.** Specifies that the stage generates the CREATE TABLE statement using information from Table name, the column definitions grid, and the values in the Create Table Properties dialog.
- **User-defined DDL.** Specifies that you enter the appropriate CREATE TABLE statement described in the section about the **SQL** tab (see [page 14](#)).

Click the button to open the **Create Table Properties** dialog to display the table space and storage expression values for generating the DDL.

- **Drop table action.** Drops the target table before it is created by the stage if **Generate DDL** is selected. This field is disabled if you choose not to create the target table. The list displays the same items as the **Create table action** list except that they apply to the DROP TABLE statement. You must have DROP TABLE privileges on your schema.
- **Enable case sensitive table/column name.** Enables the use of case-sensitive table and column names. Select to enclose table and column names in SQL statements in double quotation marks (" "). It is cleared by default.

- **Treat warning message as fatal error.** Determines the behavior of the stage when an error is encountered while writing data to a table. If the check box is selected, a warning message is logged as fatal, and the job aborts. The format of the error message is:

ORA-xxxxx Oracle error text message and row value

If the check box is cleared (the default), three warning messages are logged in the Ascential DataStage Director log file, and the job continues. The format of the error message is:

```
value of the row causing the error  
ORA-xxxxx Oracle error text message  
DBMS.CODE=ORA-xxxxx
```

The last warning message is used for Reject Link Variables. If you wish to use the Reject Link Variables functionality, you must clear the check box.

- **Pre 4.2 User-defined SQL Behavior.** Specifies the behavior of parameter marks in the WHERE clause in a user-defined SQL statement. If the check box is selected, the parameter marks for the WHERE clause in the statement always begin at :1 when the **Update action** is set to **User-defined SQL**. This is the default behavior, which is the same as that for releases of Ascential DataStage before 4.2.

If the check box is cleared, the parameter mark depends on the position of the key column. For information on writing your own SQL statements, see "[Using User-Defined SQL Statements](#)" on page 23

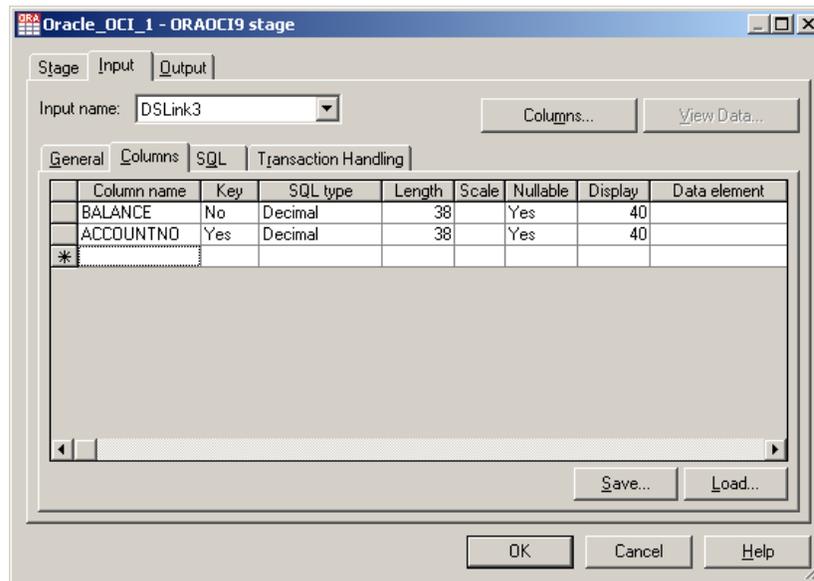
This check box appears only when you select **Delete existing rows only** and then select **User-defined SQL**.

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

Columns Tab

This tab contains the column definitions for the data written to the table.

The column definitions appear in the same order as in the Columns grid:

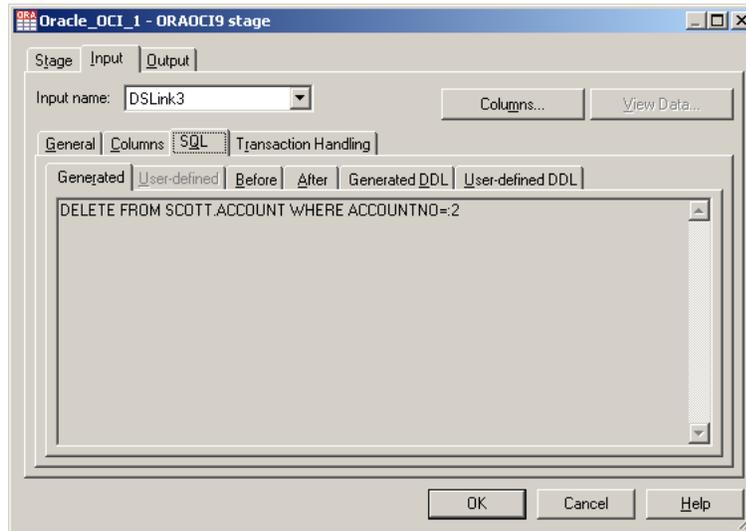


The **Columns** tab behaves the same way as the **Columns** tab in the ODBC stage. For a description of how to enter and edit column definitions, see *Ascential DataStage Designer Guide*.

SQL Tab

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

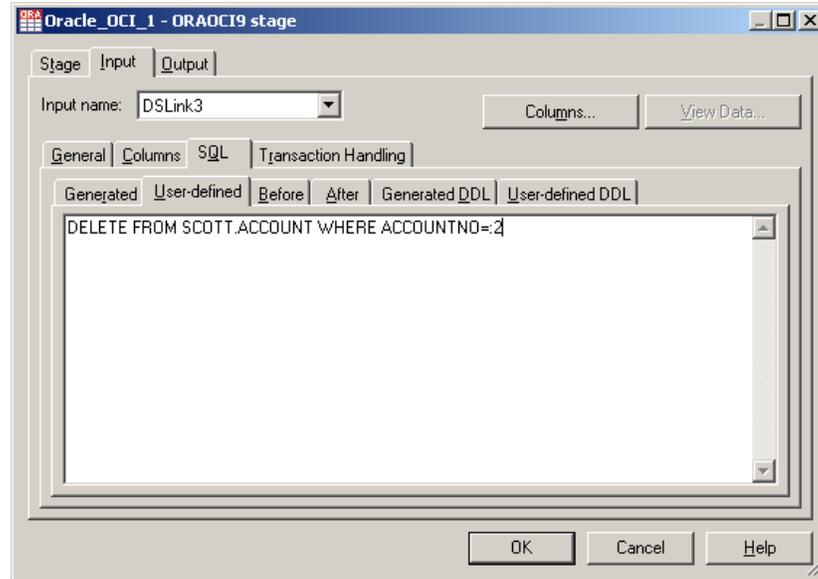
Use these tabs to display the stage-generated SQL statement and the SQL statement that you can enter.



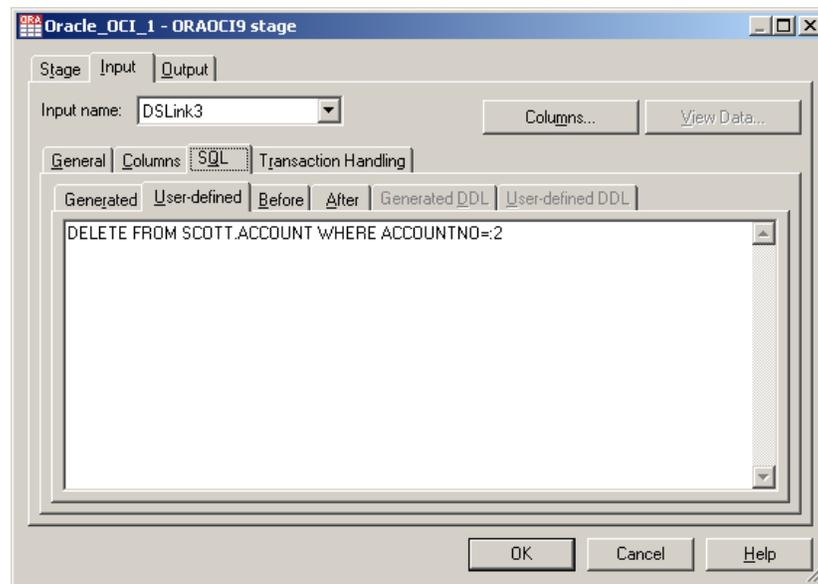
- **Generated.** This tab is displayed by default. It contains the SQL statements constructed by Ascential DataStage that are used to write data to Oracle. It is based on the current values of the stage and link properties. You cannot edit these statements, but you can use **Copy** to copy them to the Clipboard for use elsewhere, for example, user-defined SQL statements.
- **User-defined.** Contains the SQL INSERT, DELETE, or UPDATE statements executed to write data to Oracle.

The following example assumes that the **Update action** list on the **General** tab of the **Input** page is set to **User-defined SQL**.

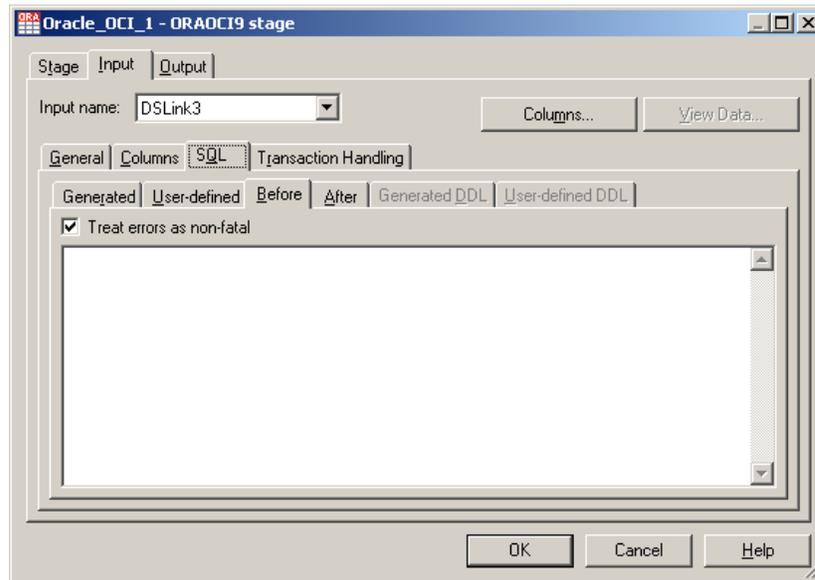
It shows the **SQL** page for DELETE SQL statement with parameters that depend on the position of the key columns. In this case, **Pre 4.2 User-defined SQL Behavior** on the **General** tab of the **Input** page is cleared. Only one parameter mark occurs for the SQL statement. The position of the key column is the important factor, as it would be for any UPDATE statements as well.



The next example assumes that the **Update action** is set to **User-defined SQL** and that the **Pre 4.2 User-defined SQL Behavior** check box on the **General** tab of the **Input** page is selected. It shows the SQL page for a DELETE SQL statement with parameters that do not depend on the position of the key columns.



- **Before.** Contains the SQL statements executed before the stage processes any job data rows.



The parameter on the **Before** tab corresponds to the Before SQL and Continue if Before SQL fails grid properties. The Continue if Before SQL fails property is represented by the **Treat errors as non-fatal** check box, and the SQL statement is entered in a resizable edit box. The **Before** and **After** tabs look alike.

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

The Before SQL is the first SQL statement to be executed. Depending on your choice, the job can continue or abort after failing to execute a Before statement. It does not affect the transaction grouping scheme. The commit or rollback is performed on a per-link basis.

Each SQL statement is executed as a separate transaction if the statement separator is a double semi-colon (;;). All SQL statements are executed in a single transaction if a semi-colon (;) is the separator.

Treat errors as non-fatal. If selected, errors caused by Before SQL are logged as warnings, and processing continues with the next command batch. Each separate execution is treated as a separate transaction. If cleared, errors are treated as fatal to the job, and result in a transaction rollback. The transaction is committed only if all statements successfully execute.

- **After.** Contains the SQL statements executed after the stage processes the job data rows. The parameters on this tab correspond to the After SQL and Continue if After SQL fails grid properties. The Continue if After SQL fails property is represented by the **Treat errors as non-fatal** check box, and the SQL statement is entered in a resizable edit box. The Before and After tabs look alike.

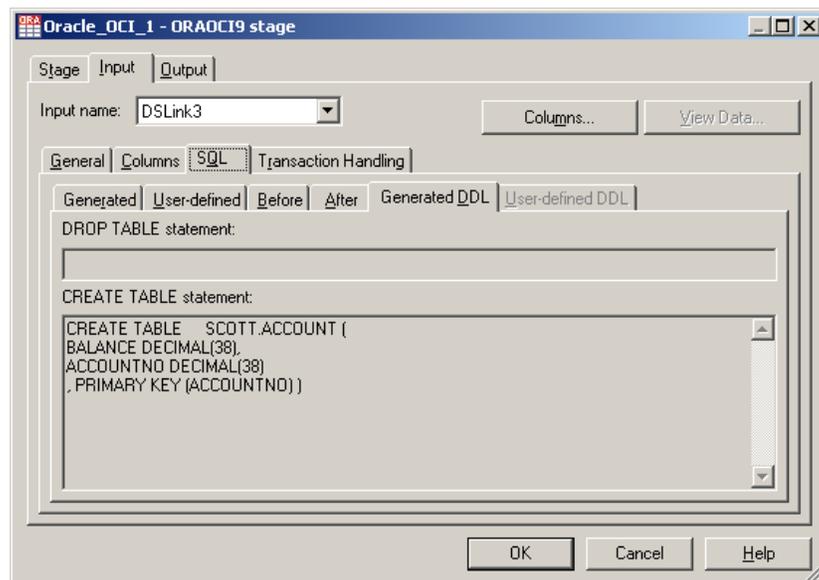
If the property value begins with FILE=, the remaining text is interpreted as a pathname, and the contents of the file supplies the property value.

The After SQL statement is the last SQL statement to be executed. Depending on your choice, the job can continue or abort after failing to execute an After SQL statement. It does not affect the transaction grouping scheme. The commit or rollback is performed on a per-link basis.

Each SQL statement is executed as a separate transaction if the statement separator is a double semi-colon (;;). All SQL statements are executed in a single transaction if a semi-colon (;) is the separator.

The behavior of Treat errors as non-fatal is the same as for Before.

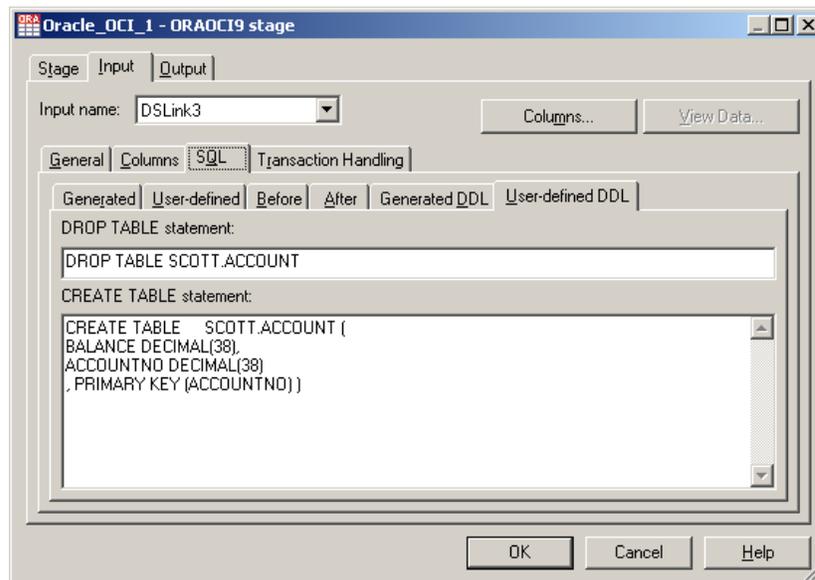
- **Generated DDL.** Select **Generate DDL** or **User-defined DDL** from the **Create table action** field on the **General** tab to enable this tab.



The **CREATE TABLE statement** field displays the CREATE TABLE statement that is generated from the column meta data definitions and the information provided on the **Create Table**

Properties dialog box. If you select an option other than **Do not drop target table** from the **Drop table action** list, the **DROP statement** field displays the generated DROP TABLE statement for dropping the target table.

- **User-defined DDL.** Select **User-defined DDL** from the **Create table action** or **Drop table action** field on the **General** tab to enable this tab. The generated DDL statement is displayed as a starting point to define a CREATE TABLE and a DROP TABLE statement. If the property value begins with FILE=, the remaining text is interpreted as a pathname, and the contents of the file supplies the property value.



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

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

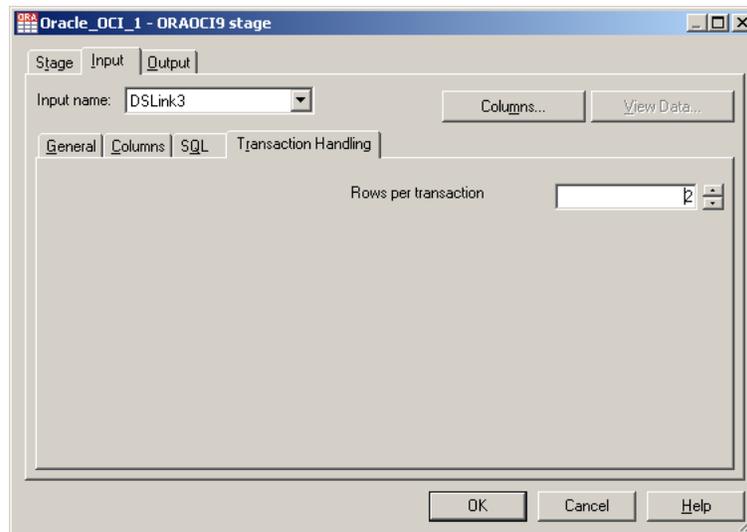
Transaction Handling Tab

Oracle OCI supports transaction grouping, that is, the grouping of input links that come from a Transformer stage. This lets you control

the group of input links for start, commit, or rollback in one transaction when writing to a single data source. You can use **On Fail** or **On Skip** values to specify whether the transaction is committed. For more information about transaction control, see *Ascential DataStage Server Job Developer's Guide*.

This tab lets you view the transaction handling features of the stage as it writes to the data source. You can choose an isolation level.

If you have a single link, the **Transaction Handling** tab looks like this.



This tab contains the following parameter:

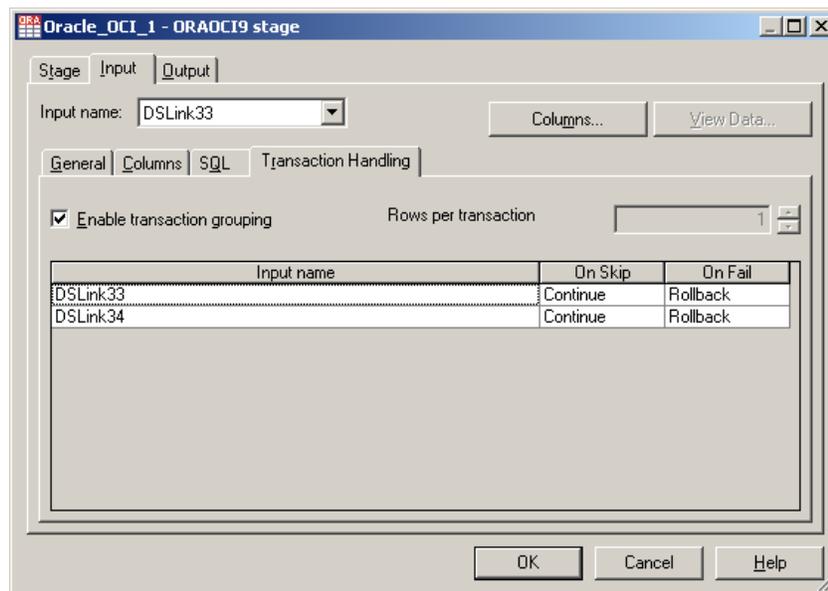
- **Rows per transaction.** If **Enable transaction grouping** is cleared, you can set **Rows per transaction** to specify the number of rows written before the data is committed to the table. The default value is 0, that is, *all* the rows are written before being committed to the table.

If you are upgrading an existing job that has a value in the **Transaction size** field on the **General** tab page, that value determines the number of rows per transaction, provided that the **Rows per transaction** field contains a value of 0.

If the **Rows per transaction** field contains a value greater than zero, this value determines the number of rows per transaction, and any value in the **Transaction size** field is ignored.

When creating a new job, use the **Rows per transaction** field to set the number of rows per transaction. Do not use the **Transaction size** field.

If you have two or more links from a single Transformer stage, the **Transaction Handling** tab looks like this.



This tab contains the following parameters:

- **Enable transaction grouping.** If selected, displays the grid with details of the transaction group to which the currently selected input link belongs. The check box is cleared by default.

If **Enable transaction grouping** is selected, a transaction group can only use a value of 1 for **Rows per transaction**. See the following section "[Handling Transactions](#)" on page 20.

- **Input name.** The non-editable name of the input link.
- **On Skip.** Specifies whether to continue or to roll back the transaction if a link is skipped because of an unsatisfied constraint on it.
- **On Fail.** Specifies whether to continue or roll back if the SQL statement fails to execute.

Handling Transactions

To specify transaction control information for a transaction group:

- 1 Click the **Transaction Handling** tab.
- 2 Select **Enable transaction grouping**.
- 3 For transaction groups, **Rows per transaction** is automatically set to 1, and you cannot change this setting.

- 4 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 non-editable name of the input link.
 - **On Skip.** Specifies whether to continue or to roll back the transaction if a link is skipped because of an unsatisfied constraint on it. Rows arriving at its link are skipped until the controlling link starts another transaction. Choose **Continue** or **Rollback** from the list.
 - **On Fail.** Specifies whether to continue or rollback if the SQL statement fails to execute. Choose **Continue** or **Rollback** from the list.

Reject Row Handling

During input link processing, rows of data may be rejected by the database for various reasons, such as unique constraint violations or data type mismatches.

The Oracle OCI stage writes the offending row to the log for the job. For the Oracle message detail, you must use the error messages returned by the Oracle database.

Ascential DataStage provides additional reject row handling. To use this capability:

- 1 Set **Array Size** to 1.
- 2 Use a Transformer stage to redirect the rejected rows.

You can then design your job by choosing an appropriate target for the rejected rows, such as a Sequential stage. Reuse this target as an input source once you resolve the issues with the offending row values.

Writing Data to Oracle

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

Using Generated SQL Statements

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

To use a generated SQL statement:

- 1 Enter a table name in the **Table name** field on the **Input** page.
- 2 Specify how you want the data to be written by choosing a suitable option from the **Update action** list. Choose one of these options for a generated statement:
 - **Clear table then insert rows**
 - **Truncate table then insert rows**
 - **Insert rows without clearing**
 - **Delete existing rows only**
 - **Replace existing rows completely**
 - **Update existing rows only**
 - **Update existing rows or insert new rows**
 - **Insert new rows or update existing rows**
 - **User-defined SQL**
 - **User-defined SQL file**

See "[Defining Input Data](#)" on page 8 for a description of each update action.

- 3 Enter an optional description of the input link in the **Description** field.
- 4 Click the **Columns** tab on the **Input** page. The **Columns** tab appears.
- 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.

- 6 Click the **SQL** tab on the **Input** page, then the **Generated** tab to view this SQL statement. You cannot edit the statement here, but you can click this tab at any time to select and copy parts of the generated statement to paste into the user-defined SQL statement.

- 7 Click **OK** to close the **ORAOCI9 Stage** dialog box. Changes are saved when you save your job design.

Using User-Defined SQL Statements

Instead of writing data using an SQL statement constructed by Ascential DataStage, you can enter your own SQL INSERT, DELETE, or UPDATE statement for each ORAOI input link. (You can include other SQL statements such as CREATE TABLE only in a Before SQL statement.) Ensure that the SQL statement contains the table name, the type of update action you want to perform, and the columns you want to write.

To enter a user-defined SQL statement:

- 1 Click the **User-defined** tab on the **SQL** tab.
- 2 Enter the SQL statement you want to use to write data to the target Oracle tables. This statement must contain the table name, the type of update action you want to perform, and the columns you want to write. Only two SQL statements are supported for input links.

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

```
insert emp (emp_no, emp_name) values (:1, :2)
```

If you specify two SQL statements, they are executed as one transaction. Do not use a trailing semicolon.

You cannot call stored procedures as there is no facility for parsing the row values as parameters.

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

- 3 Click **OK** to close the **ORAOCI9 Stage** dialog box. Changes are saved when you save your job design.

Defining Output Data

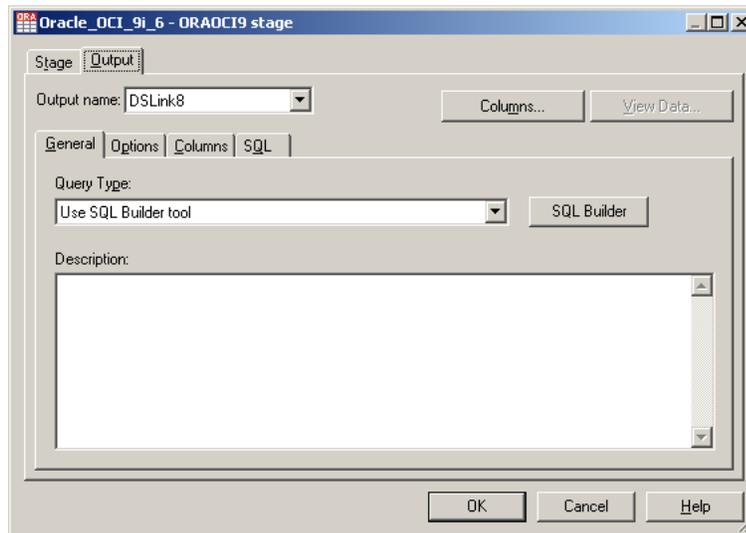
Output links specify the data you are extracting from an Oracle database. You can also specify the data on an output link using an

SQL statement constructed by Ascential DataStage or a user-defined SQL statement. These SQL statements can be:

- Fully generated, using **Use SQL Builder tool** as the **Query Type**
- Column-generated, using **Generate SELECT clause from column list; enter other clauses** as the **Query Type**
- Entered or edited entirely as text, using **Enter custom SQL statement** as the **Query Type**
- Entered from a file, using **Load SQL from a file at run time** as the **Query Type**

The SQL Builder option of fully generated SQL statements provides the most convenient method of generating SQL text. It is activated when you select **Use SQL Builder tool** as the **Query Type** (see "General Tab" on page 25). The **SQL Builder** dialog box contains all the information necessary to generate the SQL to extract data from an Oracle database.

The following sections describe the differences when you use SQL SELECT statements for generated or user-defined queries that you define on the **Output** page in the **ORAOCI9 Stage** dialog box of the GUI.



About the Output Page

The **Output** page has one field and the **General**, **Options**, **Columns**, and **SQL** 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 Oracle OCI stage.

- The **Columns...** and the **View Data...** buttons function like those on the **Input** page.

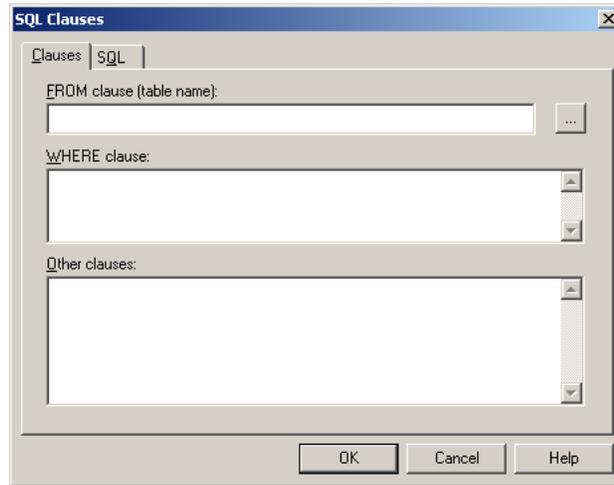
General Tab

This tab is displayed by default. It provides the type of query and, where appropriate, a button to open an associated dialog box. The **General** tab contains the following fields:

- **Query type.** Displays the following options.
 - **Use SQL Builder tool.** Specifies that the SQL statement is built using the SQL Builder graphical interface. When this option is selected, the **SQL Builder** button appears. If you click **SQL Builder**, the SQL Builder opens. See *DataStage Enterprise Edition: Parallel Job Developer's Guide* for a complete description of the SQL Builder. This is the default setting.
 - **Generate SELECT clause from column list; enter other clauses.** Specifies that DataStage generates the SELECT clause based on the columns you select on the **Columns** tab. When this option is selected, the **SQL Clauses** button appears. If you click **SQL Clauses**, the **SQL Clauses** dialog box appears (see "[SQL Clauses Dialog Box](#)" on page 26). Use this dialog box to refine the SQL statement.
 - **Enter custom SQL statement.** Specifies that a custom SQL statement is built using the **SQL** tab. See "[SQL Tab](#)" on page 30.
 - **Load SQL from a file at run time.** Specifies that the data is extracted using the SQL query in the pathname of the designated file that exists on the server. Enter the pathname for this file instead of the text for the query. With this choice, you can edit the SQL statements.
- **Description.** Lets you enter an optional description of the output link.

SQL Clauses Dialog Box

Use this dialog box to enter FROM, WHERE, or any other SQL clauses.



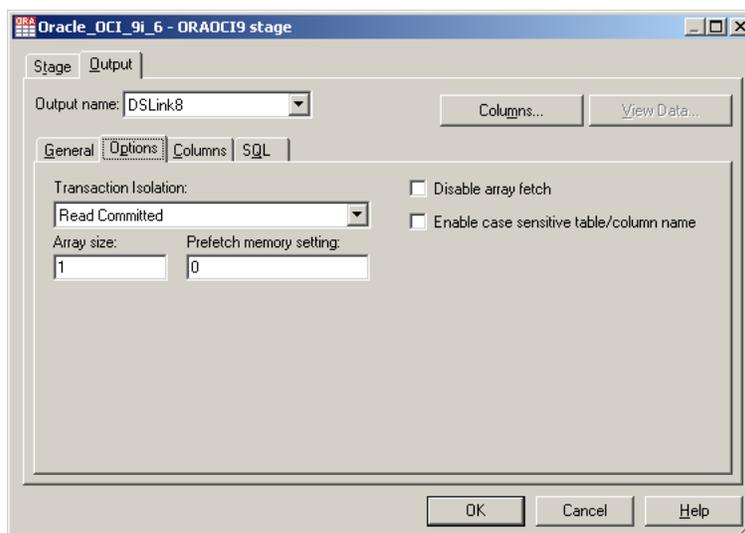
It contains the **Clauses** and **SQL** tabs.

- **Clauses tab.** Use this tab to build column-generated SQL queries. It contains optional SQL clauses for the conditional extraction of data. The Clauses tab is divided into three panes.
 - **FROM clause (table name):**. Allows you to name the table against which the SQL statement runs. To access **Table Definitions**, click ... (ellipsis).
 - **WHERE clause.** Allows you to insert an SQL WHERE clause to specify criteria that the data must meet before being selected.
 - **Other clauses.** Allows you to insert a GROUP BY, HAVING, or ORDER BY clause to sort, summarize, and aggregate data.
- **SQL Tab.** Use this tab to display the SQL statements that read data from Oracle. You cannot edit these statements, but you can use **Copy** to copy them to the Clipboard for use elsewhere.

Options Tab

Use this tab to specify transaction isolation, array size, prefetch

memory size, and case-sensitivity.



The **Options** tab contains the following parameters:

- **Transaction Isolation.** Specifies the transaction isolation levels that provide the necessary consistency and concurrency control between transactions in the job and other transactions for optimal performance.

Because Oracle does not prevent other transactions from modifying the data read by a query, that data may be changed by other transactions between two executions of the query. Thus, a transaction that executes a given query twice may experience both nonrepeatable reads and phantoms. Use one of the following transaction isolation levels:

- **Read Committed.** Takes exclusive locks on modified data and sharable locks on all other data. Each query executed by a transaction sees only data that was committed before the query (not the transaction) began. Oracle queries never read dirty, that is, uncommitted data. This is the default.
- **Serializable.** Takes exclusive locks on modified data and sharable locks on all other data. It sees only those changes committed when the transaction began plus those made by the transaction itself through INSERT, UPDATE, and DELETE statements. Serializable transactions do not experience nonrepeatable reads or phantoms.
- **Read-only.** Sees only those changes that were committed when the transaction began. This level does not allow INSERT, UPDATE, and DELETE statements.

- **Array size.** Specifies the number of rows read from the database at a time. Enter a positive integer to indicate the number of rows to prefetch in one call. This value is used both for prefetching rows and for array fetch.

Larger numbers use more memory on the client to cache the rows. This minimizes server round trips and maximizes performance by executing fewer statements. If this number is too large, the client may run out of memory.

- **Prefetch memory setting.** Sets the memory level for top-level rows to be prefetched. See Oracle documentation for further information. Express the value in number of bytes.
- **Disable array fetch.** Enables or disables Oracle array fetch. Array fetch is enabled by default. The value in **Array size** is used for array fetch size.
- **Enable case sensitive table/column name.** Enables the use of case-sensitive table and column names. Select to automatically enclose table and column names in SQL statements in double quotation marks (" "). It is cleared by default.

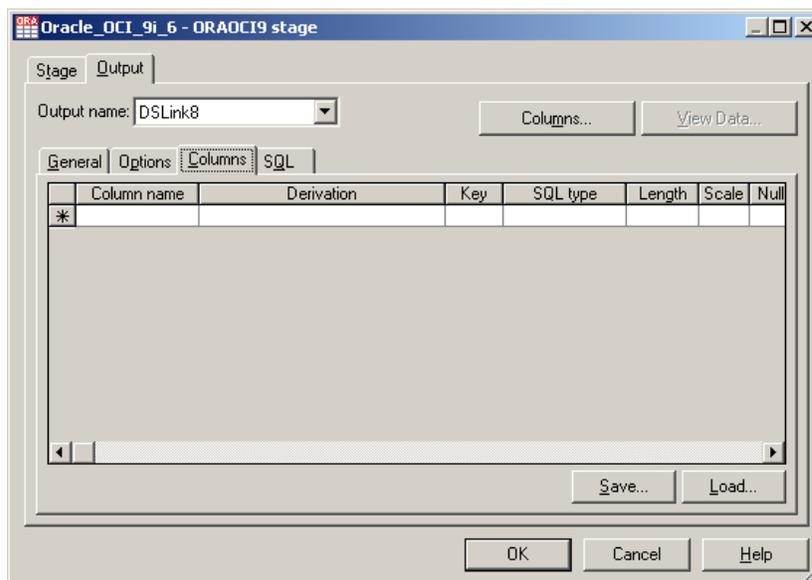
Note If **Enable case sensitive table/column name** is selected, when qualified column names are specified in the **Derivation** cell on the **Columns** tab, you must enclose these table and column names in double quotation marks (" ").

Columns Tab

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

The column tab page behaves the same way as the **Columns** tab in the ODBC stage, and it specifies which columns are aggregated. For a

description of how to enter and edit column definitions, see *Ascential DataStage Designer Guide*.



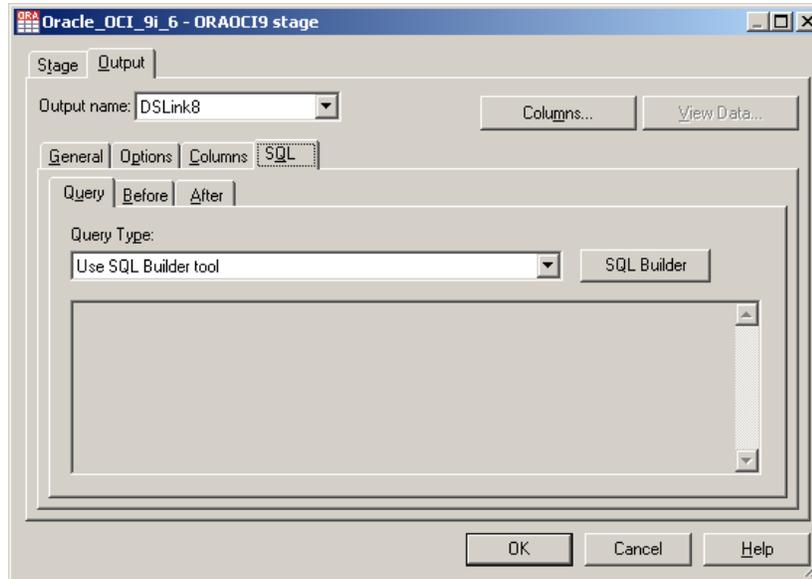
The column definitions for output links contain a key field. Key fields are used to join primary and reference inputs to a Transformer stage. For a reference output link, the Oracle OCI key reads the data by using a WHERE clause in the SQL SELECT statement. For details on how key fields are specified and used, see *Ascential DataStage Designer Guide*.

The **Derivation** cell on the **Columns** tab contains fully-qualified column names when table definitions are loaded from the Ascential DataStage Repository. If the **Derivation** cell has no value, Oracle OCI uses only the column names to generate the SELECT statement displayed in the **Generated** tab of the **SQL** tab. Otherwise, it uses the content of the **Derivation** cell. Depending on the format used in the Repository, the format is *owner.table.name.columnname* or *tablename.columnname*.

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

SQL Tab

Use this tab page to build the SQL statements used to read data from Oracle.



It contains the **Query**, **Before**, and **After** tab pages:

- **Query.** This tab is read-only if you select **Use SQL Builder tool** or **Generate SELECT clause from column list; enter other clauses** for **Query Type**. If **Query Type** is **Enter Custom SQL statement**, this tab contains the SQL statements executed to read data from Oracle. The GUI displays the stage-generated SQL statement on this tab as a starting point. However, you can enter any valid, appropriate SQL statement. If **Query Type** is **Load SQL from a file at run time**, enter the path name of the file.

- **Before.** Contains the SQL statements executed before the stage processes any job data rows. The Before is the first SQL statement to be executed, and you can specify whether the job continues or aborts after failing to execute a Before SQL statement. It does not affect the transaction grouping scheme. The commit/rollback is performed on a per-link basis.

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

- **After.** Contains the After SQL statement executed after the stage processes any job data rows. It is the last SQL statement to be executed, and you can specify whether the job continues or aborts after failing to execute an After SQL statement. It does not affect the transaction grouping scheme. The commit/rollback is performed on a per-link basis.

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

Reading Data from Oracle

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

The column definitions for reference links must contain a key field. You use key fields to join primary and reference inputs to a Transformer stage.

Oracle OCI key reads the data by using a WHERE clause in SQL SELECT statements.

Using the SQL Builder

The SQL Builder provides a graphical interface that helps you build SQL SELECT statements. These statements allow you to select rows of data from your relational database. The statement can be a simple one that selects rows from a single table, or it can be complex, performing joins between multiple tables or aggregations of values within columns.

The SQL Builder is fully documented in *DataStage Enterprise Edition: Parallel Job Developer's Guide*.

To use the SQL Builder, select **Use SQL Builder tool** from the **Query type** list on the **General** tab of the **Output** page. The **SQL Builder** button appears. Click **SQL Builder**. The SQL Builder appears.

Follow the instructions documented in *DataStage Enterprise Edition: Parallel Job Developer's Guide*.

Using Generated Queries

Ascential DataStage extracts data from an Oracle data source using a complete SQL SELECT statement that it constructs. The SQL statement is automatically constructed using the information that you entered in the stage output properties.

When you select **Generate SELECT clause from column list; enter other clauses**, data is extracted from an Oracle database using an SQL SELECT statement constructed by Ascential DataStage.

Also, the **SQL Clauses** button appears. Click **SQL Clauses**. The **SQL Clauses** dialog box appears.

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 Oracle OCI stage, the SQL SELECT statement is automatically constructed and can be viewed by clicking the **SQL** tab on the **Output** page. The SQL SELECT statement includes other appropriate clauses based on your entries in the **FROM clause (table name)**, **WHERE clause**, and **Other clauses** text boxes in the **SQL Clauses** dialog box. For example,

- Select the columns **Name, Address, City, State, AreaCode**, and **Telephone Number** from a table called Table1
- Specify the value of **AreaCode** to be 617 in the **Where clause** text box
- Specify **City** as the column to order by in the **Other clauses** text box

The SQL statement displayed on the **SQL** tab is:

```
SELECT Name, Address, City, State, AreaCode, Telephone
FROM Table1 WHERE AreaCode = 617 ORDER BY City;
```

The SELECT and FROM clauses are the minimum required. 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. Use it also to join two or more tables and limit the rows selected.
GROUP BY clause	Groups rows to summarize results.
HAVING clause	Specifies the criteria that grouped rows must meet to be selected.
ORDER BY clause	Sorts selected rows.

For more information about these clauses, see *Ascential DataStage Server Job Developer's Guide*.

Using User-Defined Queries

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

To enter a user-defined SQL query:

- 1 Select **Enter custom SQL statement** from the **Query type** list on the **General** tab on the **Output** page. The **SQL** tab is enabled.
- 2 You can edit or drag and drop the selected columns into your user-defined SQL statement. Only one SQL statement is supported for an output link. You must ensure that the table definitions for the output link are correct and represent the columns that are expected.
- 3 If your entry begins with {FILE}, the remaining text is interpreted as a pathname, and the contents of the file supplies the text for the query.
- 4 Click **OK** to close this dialog box. Changes are saved when you save your job design.

DATE Data Type Considerations

An Oracle DATE data type contains date and time information (there is no TIME data type in Oracle). Ascential DataStage maps the Oracle DATE data type to a Timestamp data type. This is the default DataStage data type when you import the Oracle meta data type of DATE.

Ascential DataStage uses a conversion of `YYYY-MM-DD HH24:MI:SS` when reading or writing an Oracle date. If the DataStage data type is Timestamp, Ascential DataStage uses the **to_date** function for this column when it generates the INSERT statement to write an Oracle date. If the DataStage data type is Timestamp or Date, Ascential DataStage uses the **to_char** function for this column when it generates the SELECT statement to read an Oracle date.

The following example creates a table with a DATE data type on an Oracle server. The imported DataStage data type is Timestamp.

```
create table dsdate (one date);
```

The results vary, depending on whether the Oracle OCI stage is used as an input or an output link:

- **Input link.** The stage generates the following SQL statement:

```
insert into dsdate(one) values(TO_DATE(:1, 'yyyy-mm-dd hh24:mi:ss'))
```

- **Output link.** The stage generates the following SQL statement:

```
select TO_CHAR(one, 'YYYY-MM-DD HH24:MI:SS') FROM dsdate
```

Oracle Data Type Support

The following tables document the support for Oracle data types. When creating DataStage table definitions for an Oracle table, specify the SQL type, length, and scale attributes as noted.

Character Data Types

The following table summarizes character data types for Oracle, their DataStage SQL type definitions, and the corresponding length attributes that you need to specify:

Oracle Data Type	Ascential DataStage SQL Type	Length	Notes
CHAR (<i>size</i>)	Char (<i>size</i>)	size	Fixed length character data of length <i>size</i> . Fixed for every row in the table (with trailing spaces). Maximum size is 255 bytes per row, default size is 1 byte per row.
VARCHAR2 (<i>size</i>)	VarChar (<i>size</i>)	size	Variable length character data. A maximum <i>size</i> must be specified. VarChar is variable for each row, up to 2000 bytes per row.

Numeric Data Types

The following table summarizes the NUMBER data type for Oracle, its DataStage SQL type definitions, and the corresponding length and scale attributes that you need to specify:

Oracle Data Type	Ascential DataStage SQL Type	Length	Scale	Notes
NUMBER (<i>p,s</i>)	Decimal Double Float Numeric Integer Real	<i>p</i> <i>p</i>	<i>s</i> <i>s</i>	<p>The DataStage SQL type definition used depends on the application of the column in the table, that is, how the column is used.</p> <p>Decimal values have a maximum precision of 38 digits. Decimal and Numeric are synonyms. The full range of Oracle NUMBER values are supported without loss of precision.</p>

Date Data Types

The following table summarizes the DATE data type for Oracle and its DataStage SQL type definition:

Oracle Data Type	Ascential DataStage SQL Type	Notes
DATE	Timestamp	<p>The default format for the default DataStage data type Timestamp is <i>YYYY-MM-DD HH24:MI:SS</i>.</p> <p>If the DataStage data type is Timestamp, Ascential DataStage uses the to_date function for this column when it generates the INSERT statement to write an Oracle date.</p> <p>If the DataStage data type is Timestamp or Date, Ascential DataStage uses the to_char function for this column when it generates the SELECT statement to read an Oracle date.</p> <p>For more information, see "DATE Data Type Considerations" on page 33</p>

Miscellaneous Data Types

The following table summarizes miscellaneous data types for Oracle and its DataStage SQL type definition:

Oracle Data Types	Ascential DataStage SQL Type	Notes
CLOB	SQL_LONGVARCHAR	The Oracle OCI Plug-in supports the CLOB data type by mapping the LONGVARCHAR data type with a precision greater than 4 KB to Oracle's CLOB data type. To work with a CLOB column definition, choose DataStage's LONGVARCHAR as the column's data type and provide a Length of more than 4 KB in the Columns tab. The maximum size supported by DataStage is 2 GB. A column with a data type of CLOB cannot be used as a key.

For a list of unsupported Oracle data types, see "[Functionality](#)" on page 2.

Handling \$ and # Characters

Ascential DataStage has been modified to enable it to handle Oracle OCI 9i or 10g databases which use the DataStage reserved characters # and \$ in column names. Ascential DataStage converts these characters into an internal format and then converts them back as necessary.

To take advantage of this facility, do the following:

- In Ascential DataStage Administrator, open the **Environment Variables** dialog box for the project in question, and set the environment variable DS_ENABLE_RESERVED_CHAR_CONVERT to true (this can be found in the General\Customize branch).
- Avoid using the strings __035__ and __036__ in your Oracle column names (these are used as the internal representations of # and \$ respectively).

Import meta data using the Plug-in Meta Data Import tool; avoid hand-editing (this minimizes the risk of mistakes or confusion).

Once the table definition is loaded, the internal column names are displayed rather than the original Oracle names both in table definitions and in the Data Browser. They are also used in derivations and expressions. The original names (that is, those containing the \$ or

#) are used in generated SQL statements, however, and you should use them if entering SQL in the job yourself.

When using an Oracle OCI 8i Plug-In in a server job, you should use the external names when entering SQL statements that contain Oracle columns. The columns within the stage are represented by :1, ;2, etc. (parameter markers) and bound to the Oracle columns by order, so you do not need to worry about entering names for them. This applies to:

- Query
- Update
- Insert
- Key
- Select
- Where clause

For example, for an update you might enter:

```
UPDATE tablename SET ##B$ = :1 WHERE $A# = :2
```

Particularly note the key in this statement ($\$A\#$) is specified using the external name.

Index

A

array fetch 28
array size 26
 specifying 10, 28

C

case-sensitive table or column names 11, 27, 28
character data types 34
character set mapping 5, 6, 7
clearing tables 9, 22
CLOB data type 36
column-generated SQL queries. See generated SQL queries.
configuration requirements 4
connecting to Oracle databases 6
converting projects 7
CREATE TABLE statement 11, 17, 23
creating tables 9, 11, 17–18, 33

D

Data Browser 9
data types
 character 34
 CLOB 36
 DATE 33, 35
 numeric 35
 support for 34
DATE data type 33, 35
defining
 character set mapping 6, 7
 OCI connections 5–7
 OCI input data 8–21
 OCI output data 23–29
DELETE statement 14, 21, 22, 23

dialog boxes
 ORAOCI9 Stage 6, 8, 23, 24
 SQL Clauses 25, 26, 32
dollar sign (\$) character 36
DROP TABLE statement 11, 18
dropping tables 11

E

editing an ORAOCI9 stage 6
error handling 12

F

fatal errors 12
FROM clause 26, 32
functionality 2

G

generated SQL queries 25, 26, 30, 31
generating SQL statements
 for reading data 31
 for writing data 11, 14, 22
GROUP BY clause 26, 32

H

handling
 errors 12
 rejected rows 21
HAVING clause 26, 32

I

input link 2, 8, 9, 12, 18, 20, 23, 33
Input page 5, 8–21
 table name for 9
 update action for 9
INSERT statement 14, 21, 22, 23

N

numeric data types 35

O

Oracle database, connecting to 6
ORAOCI9 Stage dialog box 6, 8, 23, 24
ORDER BY clause 26, 32
output link 2, 5, 23, 29, 33
Output page 5, 24, 24–29

P

pound (#) character 36
prefetch memory size 27, 28
prefetching rows 28

R

reading data from Oracle 31–33
reject row handling 21
Repository 9

S

SELECT clause 32
SELECT statement 24, 29, 31
special characters 36
SQL Builder 24, 25, 31
SQL Clauses dialog box 25, 26, 32
SQL queries
 defined at run time 25
 generated 25, 26, 30, 31
 in file 33
 SQL Builder 25
 user-defined 25, 33
SQL statements
 DELETE 23
 examples 23, 32, 33
 FROM clause 26, 32
 GROUP BY clause 26, 32
 HAVING clause 26, 32
 INSERT 23
 ORDER BY clause 26, 32
 SELECT 24, 29, 31
 syntax 32
 UPDATE 23
 WHERE clause 26, 29, 32
SQL, user-defined 9, 10, 14, 23, 25, 33
Stage page 5, 7

T

table name 9
tables
 clearing 9, 22
 creating 9, 11, 17–18, 33
 reading from 31–33
 writing to 21–23
transaction grouping 2, 10, 16, 17, 18, 20
transaction handling 18, 20
transaction isolation 26

U

update action, input pages 9
UPDATE statement 14, 21, 22, 23
user-defined SQL 9, 10, 14, 23, 25, 33
using
 generated SQL statements
 writing data to Oracle 22
 user-defined SQL statements
 writing data to Oracle 23

W

warning messages 12
WHERE clause 26, 29, 32
writing data to Oracle 21–23