# **ActiveX Script Task Properties**

Use this dialog box to specify the code that will perform the functions you need to customize your Data Transformation Services (DTS) package (for example, skipping a row of source data that contains invalid data).

## **Options**

### Description

Specify a description for the Microsoft® ActiveX® Script task. This description becomes the label for the task icon placed on the DTS Designer design sheet.

### Language tab

This tab lets you specify the scripting language and the functions to use in the ActiveX script.

### Language

Select an available scripting language. When you install scripting languages on the computer, this list will update automatically. Microsoft Visual Basic® Scripting Edition (VBScript) and Microsoft JScript® are available by default.

### Functions

Select a function from the script language library to be placed into the **ActiveX script text box**. Double-clicking on a function name inserts the function code into the text box at the position of the cursor.

### **Entry Function**

Specify the name of the function that will be the entry point when the script runs. Only one function can be specified as an entry point for an ActiveX Script task.

### **Browser tab**

View the tree that contains the return code constants, the package global variables, and the lookups available for use in the ActiveX script. To copy

information into the **ActiveX script text box**, expand the nodes and doubleclick on an item.

## ActiveX script text box

Write or paste the scripting code necessary to perform the functions you need. The scripting editor has limited functionality. It does not include features such as statement completion or color-coding of reserved words. Comment lines use the apostrophe (') character syntax in any column, and all text from the comment character to the end of the line is ignored.

## Auto Gen.

Generate a single function and a line of code containing the return code, if supported by the selected scripting language.

**IMPORTANT** If there is scripting code in the **ActiveX script text box**, the code will be deleted when you click **Auto Gen**.

## Browse

Display the **Select File** dialog box, where you can select a file containing code. When you select a file, the contents of that file are copied into the **ActiveX script text box**.

**IMPORTANT** Opening an external script file from the **Select File** dialog box will overwrite any existing code in the **ActiveX script text box**. Therefore, always select code files before inserting any other code into the box.

### Parse

Check the code for syntax errors.

### Save

Display the **Save As** dialog box, where you can save all the code in the ActiveX script text box into a file on the local hard drive or on any mapped drive.

### Undo

Reverse a limited number of text entry actions. You cannot undo actions such as adding code through the **Select File** dialog box.

# See Also

Using ActiveX Scripts in DTS Using Global Variables with DTS Packages Debugging ActiveX Scripts Using Return Codes in DTS ActiveScriptTask Object

# **ActiveX Script Transformation Properties**

Use this dialog box to define a transformation with a Microsoft® ActiveX® script. In this use of an ActiveX script, transformations are executed for every row of source data.

## **Options**

### Language tab

This tab lets you specify the scripting language and the functions to use in the ActiveX script.

### Language

Select an available scripting language. When you install scripting languages on the computer, this list will update automatically. Microsoft Visual Basic® Scripting Edition (VBScript) and Microsoft JScript® are available by default.

### Functions

Select a function from the script language library to be placed into the **ActiveX script text box**. Double-clicking on a function name inserts the function code into the text box at the position of the cursor.

### **Entry function**

Specify the name of the function that will be the entry point when the ActiveX script runs. If you are adding functions for data pump phases, this function only refers to the entry point for the Row Transform phase function. If you are not adding a function for the Row Transform phase, this option is disabled.

### **Browser tab**

View the tree that contains the return code constants, the Data Transformation Services (DTS) package global variables, the source and destination columns, and the lookups available for use in the ActiveX script. Click to expand nodes, and double-click on an item to copy the information into the **ActiveX script text box**.

## Phases tab

Select the data pump phases for which you plan to add ActiveX scripts, and then type the name of each function beneath the appropriate phase name. You must include an ActiveX script function in the text box for each selected phase, and the function name in the scripting box must match the function name entered on this tab.

**Note** This tab only appears if you enabled multiphase data pump functionality in SQL Server Enterprise Manager.

## ActiveX script text box

Write or paste the scripting code necessary to perform the functions you need. The scripting editor has limited functionality. It does not include features such as statement completion or color-coding of reserved words. Comment lines use the apostrophe (') character syntax in any column, and all text from the comment character to the end of the line is ignored.

## Auto Gen.

Generate a single function and a line of code containing the return code, if supported by the selected scripting language.

**IMPORTANT** If there is scripting code in the **ActiveX script text box**, the code will be deleted when you click **Auto Gen**.

## Browse

Display the **Select File** dialog box, where you can select a file containing code. When you select a file, the contents of that file are copied into the **ActiveX script text box**.

**IMPORTANT** Opening an external script file from the **Select File** dialog box will overwrite any existing code in the **ActiveX script text box**. Therefore, always select code files before inserting any other code into the box.

Parse

Check the syntax of the ActiveX script.

### Save

Display the **Save As** dialog box, where you can save all the code in the **ActiveX script text box** into a file on the local hard drive or on any mapped drive.

### Undo

Reverse a limited number of text entry actions. You cannot undo actions such as adding code through the **Select File** dialog box.

### Test

Test the transformation by executing it against a part of the source data and copying the results to a temporary text file for preview purposes. This option is not available for ActiveX script transformations in a Data Driven Query task.

Because test mode sends data to a file rather than the actual destination, problems with the provider, the destination, or data type overflow may not be detected.

## See Also

Using ActiveX Scripts in DTS

Mapping Column Transformations

Using Global Variables with DTS Packages

**Debugging ActiveX Scripts** 

Using Return Codes in DTS

Multiphase Data Pump Functionality

# **Add/Edit Assignment**

Use this dialog box to specify the source for the property whose value will be dynamically assigned.

## **Options**

### Source

Select an available source from the list. The source for the dynamic property value can be an .ini file, an SQL query, a global variable, an environment variable, a constant, or a data file.

### **Source information**

Further specify the source information required to extract the dynamic property value. Depending on which **Source** you select, different options become available.

### File

For an **INI File** or **Data File** source, select the file from which you want the property value extracted.

### Section

For an **INI File**, select the section of the file containing the property value from the list.

### Key

For an **INI File**, select the key containing the property value from the list.

### Preview

View the current value of the property to be dynamically assigned.

### Refresh

Redisplay the property value displayed in the **Preview** box based on the most current information.

### Connection

Select an available Data Transformation Services (DTS) connection from the list.

### Query

Type a SQL query whose result set will be dynamically assigned to the selected property. When you assign the results of a query to a DTS package property, the Dynamic Properties task uses only the results of the first column of the first row.

### Parse

Check the syntax of the SQL query.

### Browse

Display the **Select File** dialog box, where you can select an .sql file containing a SQL query to enter in the **Query** box.

### Variable

Select a global variable or environmental variable from the list.

### **Create Global Variables**

Create a global variable available for selection as a dynamic properties assignment.

### Constant

Type a constant value to be assigned to the selected property.

## Select

Select a property value from a list of DTS constants.

## See Also

**Dynamic Properties Task** 

DynamicPropertiesTask Object

Using Global Variables with DTS Packages

# **Advanced Copy Options**

Use this dialog box to specify which objects to transfer from one Microsoft® SQL Server<sup>TM</sup> database to another.

## **Options**

### Transfer database users and database roles

Transfer all database users and roles.

## Transfer SQL Server logins (Windows and SQL Server logins)

Transfer all SQL Server logins.

## **Transfer object-level permissions**

Transfer all object-level permissions.

### **Transfer indexes**

Transfer indexes for all tables transferred, if applicable.

## **Transfer triggers**

Transfer triggers for all tables transferred, if applicable.

## Transfer full text indexes

Transfer full-text indexes on all tables transferred, if applicable.

## **Transfer PRIMARY and FOREIGN keys**

Transfer PRIMARY and FOREIGN key definitions for all tables transferred.

## **Generate Scripts in Unicode**

Transfer data as **Unicode**. Useful if source data contains **nchar** data types. For more information about Unicode, see <u>Unicode Data</u>.

## Use quoted identifiers when transferring objects

Enclose all object names in quotation marks.

# See Also

Copy SQL Server Objects Task

Transfer Object

# **Advanced Connection Properties**

Use this dialog box to set custom values for certain OLE DB initialization properties. The OLE DB advanced properties available for customization will vary depending on the properties supported by a particular provider.

To change an OLE DB property value, in **Value**, click a cell, and then type the new value.

**Note** If you need to work more closely at the OLE DB level, use the OLE DB Rowset Viewer, available with the OLE DB Software Development Kit (SDK). The OLE DB Rowset Viewer offers a simple way to view and manipulate OLE DB rowsets with the added ability to call and manipulate other OLE DB methods from the data source, session, command, rowset, transaction, and notification objects supported by any OLE DB provider.

## See Also

DTS Connections

# **Bulk Insert Task Properties (General Tab)**

Use this tab to specify the properties that will direct the Bulk Insert task to import data from a data file and copy it to the specified Microsoft® SQL Server<sup>TM</sup> table or view.

The parameters in the Bulk Insert properties map to most of the parameters in the Bulk Insert command. For more information about any of the options in the user interface, see <u>BULK INSERT</u>.

# Options

### Description

Describe the Bulk Insert task. This description becomes the label for the task placed on the Data Transformation Services (DTS) Designer design sheet.

### **Existing connection**

Choose the SQL Server connection that specifies the appropriate destination database for the data.

### **Destination table**

Specify the SQL Server table or view in the database to which the data is to be copied.

### Refresh

Populate the database list on Microsoft Windows® 98 computers.

## Source data file

Specify the name of the file containing the data to be copied. Click the browse (...) button to search for the source data file.

## Use format file

Specify the full path and name of the format file. The format file describes the contents of the data file, which has been created using the **bcp** utility. Click the browse (...) button to search for the source data file.

### **Specify format**

Specify the row and column delimiters in the source data file to ensure that the file will be parsed correctly when it is read.

### **Row delimiter**

Specify the delimiter used to denote the end of a row for each line of data in the source file. A newline delimiter **{LF}** is used by default.

### **Column delimiter**

Specify the delimiter used to denote the end of a column for each row in the source file. A tab is used by default.

### Generate

Display the **Select a data file and a format file** dialog box, where you search for the source data file and its associated format file on the local computer, as well as on all mapped network drivers. Displays a preview of the data before the Bulk Insert task is saved or executed.

## See Also

Bulk Insert Task BulkInsertTask Object

# **Bulk Insert Task Properties (Options Tab)**

Use this tab to specify properties with which to organize your file.

### **Check constraints**

Ensure that any constraints on the destination table are checked during the bulk copy operation. By default, constraints are ignored.

### **Enable identity insert**

Specify that there are values in the data file for an identity column.

### Sorted data

Indicate to the task that the data in the data file has been sorted on the specified column. The column name that you supply must be a valid column in the destination table.

### **Keep NULL values**

Specify that any columns containing a null value should be retained as null values, even if a default value was specified for that column in the destination table.

### Lock entire table

Specify a table-level lock on the destination table for the duration of the BULK INSERT operation.

### Code page

Indicate to the task that your data file has **char**, **varchar**, or **text** columns with character values greater than 127 or less than 32, and then choose the appropriate code page value so that the data can be parsed correctly.

### Data file type

Specify the type of data in the data file.

### **Insert batch size**

Specify the number of rows in a batch. The default is the entire data file.

The following values for the **Batch size** property have these effects:

If you set **Batch size** to zero, the data is loaded in a single batch. The first row that fails will cause the entire load to be canceled, and the step fails.

If you set **Batch size** to one, the data is loaded a row at a time. Each row that fails is counted as one row failure. Previously loaded rows are either committed or, if the step has joined the package transaction, provisionally retained in the transaction, subject to later commit or rollback.

If you set **Batch size** to a value greater than one, the data is loaded one batch at a time. Any row that fails in a batch fails that entire batch; loading stops, and the step fails. Rows in previously loaded batches are either committed or, if the step has joined the package transaction, provisionally retained in the transaction, subject to later commit or rollback.

### **Only copy selected rows**

Specify whether a continuous range of rows should be copied in a range.

### Starting with row

Specify the first row from which to start copying.

## Stopping at row

Specify the last row to copy.

## See Also

BULK INSERT

Bulk Insert Task

BulkInsertTask Object

# **Calendar Names**

Use this dialog box to:

- Change the strings representing the different months and days of the week, and A.M. and P.M. representations.
- Change the language used to represent the date and time data.

## **Options**

### Name

Identify the month, day, or A.M. and P.M. designation.

### Value

View or change the current value for the month, day, or A.M. and P.M. designation.

### Language

Select a language to use for date and time data.

### **Set Language Defaults**

Set the default language for the data used in the Date Time String transformation. Click this option after selecting from the **Language** list to set this language as the default.

## See Also

**Date Time String Transformation** 

**Transformation Types** 

Mapping Column Transformations

# **Column Order**

Use this dialog box to view the names of the source and destination columns mapped in the transformation, and to change the source and destination columns included. You may need to change the order of the column mappings in transformations where there is an unequal number of source and destination columns (for example, where a single source column is copied to multiple destination columns).

## **Options**

## **Columns text box**

Edit column mappings by clicking a **Source** cell or **Destination** cell, selecting a column name or **<ignore>** from the list, and repeating the process for any additional **Source** and **Destination** cells.

## See Also

Mapping Column Transformations Copy Column Transformation Transformation Types

# **Confirm Package Owner Password**

Use this dialog box to confirm owner passwords.

# Options

## **Password text box**

Validate the owner password you typed in the **Save DTS Package** dialog box by retyping the password. If you have set an owner password, a Data Transformation Services (DTS) package user needs this password to edit or run the package.

## See Also

Handling Package Security in DTS

# **Confirm Package User Password**

Use this dialog box to confirm Data Transformation Services (DTS) package user passwords.

# **Options**

### **Password text box**

Validate the user password you typed in the **Save DTS Package** dialog box by retyping the password. Package users with access only to the user password can run the package. However, they can neither open nor edit the package unless they also have access to the owner password.

## See Also

Handling Package Security in DTS

# **Connection Properties**

Use this dialog box to add a connection to a data source, which can be either a new connection or a copy of an existing connection. Not all of the following options are available for all providers. A subset of the options will be available, depending on the provider selected.

## **Options**

### **New connection**

Name a new connection to a data source.

### **Existing connection**

Select an existing connection to a data source.

### **Data source**

Select an OLE DB provider from the list of available providers. By default, the Microsoft® OLE DB Provider for SQL Server is selected. If your data source or data destination is an instance of Microsoft SQL Server<sup>TM</sup>, it is recommended that you use this provider, because it offers options such as high-performance bulk loading of data and setting batch size when copying and transforming data.

### File Name

Specify the database path and file name holding the data to be imported (for example, C:\MyData.xls, \\Sales\Database\Northwind.mdb).

### Username

Specify a user name for the database connection.

### Password

Specify a password for the database connection.

### **UDL Filename**

Specify the name of the data link (.udl) file that contains the connection

string.

## Always read properties from UDL file

Specify that the package resolve information in the data link (.udl) file at run time. If you do not select this check box, the connection string is copied from the .udl into the package, and the file is not referenced again. Connection changes then can be modified only by editing the Data Transformation Services (DTS) package directly.

## **Properties**

Display the **Data Link Properties** dialog box. Changes made in the dialog box will be incorporated into the package created during the current session and will not change the data link file.

### **User/System DSN**

Specify the name of the existing user or system data source name (DSN) that points to the data source.

### New

Display the **Create New Data Source** dialog box to create an ODBC DSN. For more information about creating an ODBC data source, search in the Platform SDK section in the MSDN® Library at <u>Microsoft Web site</u>.

### File DSN

Specify the name of the existing file DSN that points to the data source.

### Server

Specify the name of the server holding the data source.

## **Use Windows Authentication**

Specify that the package use Windows Authentication for login to an instance of SQL Server.

## **Use SQL Server Authentication**

Specify that the package use SQL Server Authentication for login to an instance of SQL Server.

## Database

List databases on the specified instance of SQL Server.

## Refresh

Cause the database list to populate on Microsoft Windows® 98 computers.

## Advanced

Display the **Advanced Connection Properties** dialog box, where you can enter custom settings. For more information about the OLE DB connection properties, search in the Platform SDK section in the MSDN Library at <u>Microsoft Web site</u>.

## See Also

**DTS Connections** 

Data Link Connection

Transform Data Task

# **Copy SQL Server Objects Task (Source Tab)**

Use this tab to select the source of the data to be copied and to provide the appropriate security context through which users can access the database.

## **Options**

### Description

Specify an optional description of the Copy SQL Server Objects task. This text becomes the name of the Copy SQL Server Objects task icon on the Data Transformation Services (DTS) Designer design sheet.

### Server

Specify the name of the server containing the data source.

### **Use Windows Authentication**

Specify that the package use Windows Authentication for login to the Microsoft® SQL Server<sup>™</sup> database.

### **Use SQL Server Authentication**

Specify that the package use SQL Server Authentication for login to the SQL Server database.

### User name

Specify a user name for the database connection.

### Password

Specify a password for the database connection.

### Database

List databases on the specified server.

### Refresh

Cause the database list to populate on computers running Microsoft Windows® 98.
# See Also

Copy SQL Server Objects Task

TransferObjectsTask Object

# **Copy SQL Server Objects Task (Destination Tab)**

Use this tab to specify the destination to which the data is to be copied and to provide the appropriate security context through which users can access the database.

### Server

Specify the name of the server containing the data source.

## **Use Windows Authentication**

Specify that the package use Windows Authentication for login to the Microsoft® SQL Server<sup>™</sup> database.

## **Use SQL Server Authentication**

Specify that the package use SQL Server Authentication for login to the SQL Server database.

### User name

Specify a user name for the database connection.

### Password

Specify a password for the database connection.

## Database

List databases on the specified server.

### Refresh

Cause the database list to populate on Microsoft Windows® 98 computers.

# See Also

Copy SQL Server Objects Task

# **Copy SQL Server Objects Task (Copy Tab)**

Use this tab to specify which objects to transfer from one Microsoft® SQL Server<sup>™</sup> database to another. You can transfer only from one instance of SQL Server version 7.0 to another, from an instance of SQL Server 7.0 to an instance of SQL Server 2000, and from one instance of SQL Server 2000 to another.

#### **Create destination objects**

Create destination objects for all objects to be transferred (tables, views, stored procedures, defaults, rules, constraints, user-defined data types, logins, users, roles, and indexes). Specify the following copy options:

#### **Drop destination objects first**

Drop all corresponding destination objects before creating new ones.

### Include all dependent objects

Include all dependent objects, such as the tables supporting a view, in the transfer of data.

#### Copy data

Enable the copying of SQL Server data from source to destination. Specify the following copy options:

#### **Replace existing data**

Overwrite existing data in the destination objects with the new data from the specified source.

#### Append data

Retain existing data in the destination object, and append new data from the specified source.

#### **Use Collation**

Enable the copying of data between different collations. For more information on using different collations with Data Transformation Services (DTS), see <u>Data Conversion and Transformation Considerations</u>.

# **Copy all objects**

Transfer all objects associated with the specified data source.

## **Select objects**

Display the **Select Objects** dialog box, where you can select or remove objects from the transfer process. Clear the **Transfer all objects** check box to enable this option.

## Use default options

Set the advanced transfer options to their defaults.

## **Options**

Display the **Advanced Copy Options** dialog box, where you can select or remove the specific objects to be transferred. Clear the **Use default options** check box to enable this option.

# Script file directory

Specify the directory to which the script file and log files are written. The script file directory must exist on the computer on which the task runs.

# See Also

Copy SQL Server Objects Task

TransferObjectsTask2 Object

# **Create Binding Table**

Use this dialog box to write your own SQL CREATE TABLE statement for the binding table used by the parameterized queries, or to customize the existing binding table. By default, Data Transformation Services (DTS) Designer displays a CREATE TABLE statement for the selected source table. You can edit the table name or any column definitions; however, use care because the new binding table is created immediately in the destination.

# See Also

Building a Data Driven Query

# **Create Database**

Use this dialog box to define a new database for a data source connection. You can define a new database only when connecting to an instance of Microsoft® SQL Server<sup>™</sup> through the Microsoft OLE DB Provider for SQL Server or through the Microsoft OLE DB Provider for ODBC. The database files will be put into the same location as your **master** database files. If you are unable to create a new database, make sure your login has the appropriate permissions.

This dialog box is provided for simplicity and convenience, but does not include all available options for creating databases. To have access to all options, click **New Database** in SQL Server Enterprise Manager.

Make sure to follow SQL Server naming conventions for the database, and set the data file size and log file size for the new database appropriately. For more information, see <u>Naming Conventions for Instances of SQL Server 2000</u>.

# See Also

Creating a Database

# **Create Destination Table**

Use this dialog box to write your own SQL CREATE TABLE statement for the destination table or to customize the existing destination table. By default, Data Transformation Services (DTS) Designer displays a CREATE TABLE statement for the selected source table. You can edit the table name or any column definitions; however, use care because the new table is created immediately in the destination database.

# **Create New Transformation**

Use this dialog box to select a transformation from the list to apply to the source and destination columns. The list includes all transformation types supplied with Data Transformation Services (DTS), as well as any custom transformations that were added and registered.

The most frequently used transformation types are **Copy Column** and **ActiveX Script**. Use **Copy Column** when you simply want to copy data from a source to a destination without transforming the data. Use **ActiveX Script** when you want to use scripting code to transform the data.

# See Also

Mapping Column Transformations

**Transformation Types** 

# **Custom Task Properties**

Use this dialog box to view the properties of a Data Transformation Services (DTS) custom task. All properties are listed, though not all of the listed properties can be edited (for example, BLOB data such as an image is unavailable to be edited in this dialog box). If a property can be edited, click in the cell under the **Value** column and then type the new value.

# See Also

DTS Custom Task Building a DTS Custom Task CustomTask Property CustomTask Object CustomTaskUI Object

# **Custom Transformation Properties**

Use this dialog box to view and edit the properties of the custom transformation. This dialog box appears only if you loaded a custom transformation into Data Transformation Services (DTS) Designer and did not write your own user interface for the transformation.

The properties listed in the dialog box are specific to how the transformation was written, as is the availability of the properties for editing. To edit a custom transformation property value, click on the **Value** cell for that row in the table and type the new value.

# See Also

**Building a DTS Custom Transformation** 

# **Data Driven Query Task Properties (Source Tab)**

Use this tab to select a table or view, or the results of a query as a data source for the data driven queries.

# **Options**

## Description

Type a description of this Data Driven Query task.

# Connection

Select an existing connection to a data source from the list.

# **Table/View**

Select a table or view from the data source specified on the connection. There is no guaranteed ordering of rows.

# SQL query

Specify that an SQL statement retrieves the data from the data source. Type the SQL statement.

# Parameters

Map global variables to input parameters. This option can only be performed after specifying an SQL query that includes parameters (using one or more question marks as a parameter placeholders).

# Preview

View the data in the selected source table or view.

**Note** If your source connection is a dBase data source and you specify an SQL query, the source data may not be available for viewing.

# **Build query**

Use the Data Transformation Services (DTS) Query Designer to create the SQL statement to execute against the data source.

## Browse

Specify the location of an SQL query to execute.

# **Parse query**

Check the syntax of the SQL statement prior to execution.

# See Also

Data Driven Query Task

Using Parameterized Queries in DTS

# **Data Driven Query Task Properties (Bindings Tab)**

Use this tab to configure the bindings to which the parameters of the data driven queries will map. If you intend to use multiple queries in a Data Driven Query task, the binding table must include columns for all parameters in the queries.

# **Options**

### Connection

Select an existing data source connection to use as the basis for the parameter binding table.

### Table name

Select an existing table to use for the parameter bindings. Make sure the table includes columns for all parameters in the queries you will define.

### Create

Create a new binding table to use for the parameters. A binding table must exist before you can specify transformations.

### Name

View the column name.

## Туре

View the destination native data type using OLE DB data type mappings.

## Nullability

View whether binding columns allow null values. The default is true.

## Size

View the width of binding columns where applicable.

## Precision

View the precision of binding columns where applicable.

### Scale

View the number of digits in scale where applicable.

# See Also

Data Driven Query Task

# **Data Driven Query Task Properties (Transformations Tab)**

Use this tab to select and graphically map the source and binding columns composing each transformation. Graphically mapping a transformation is not required. Instead, you may specify the source and destination columns for a transformation using selection boxes in the **Transformation Options** dialog box.

Column mappings are represented on this tab by arrows connecting columns in the **Source** and **Binding table** boxes. By default, the Data Driven Query task maps an equal number of matched source and binding table columns using a single many-to-many mapping.

**IMPORTANT** Unless you are an advanced user or have specialized needs, you should not need to change the default column mappings for a Data Driven Query task.

# Options

# **Phases filter**

Select the data pump phase for which you want to add a transformation. This advanced option is only available if the **Show multi-phase pump in DTS Designer** check box is selected in SQL Server Enterprise Manager. The default is **Row transform** phase.

# Name

View the name of the selected transformation. You must click on a mapping line to display the name of the transformation. Otherwise **<none selected>** is displayed. By default, the transformations are named **DTS\_Transformation\_***n*, where *n* is equal to the ordinal position of the

mapped transformation.

# Туре

View the type of column-level transformation for the selected mapping. You must click on a mapping line to display the transformation type. Otherwise

<**none selected**> is displayed. If you intend to create a new transformation, this value is ignored.

#### New

Display the **Create New Transform** dialog box, where you select a type of column-level transformation.

### Edit

Display the **Transformation Options** dialog box, where you can modify an existing transformation. Select the mapping line of the transformation you want to modify before clicking **Edit**.

### Delete

Delete a selected transformation. Select the mapping line representing the transformation you want to remove from the task before clicking **Delete**.

#### Source

View the columns of the source data in graphical form, and click one or more columns to include in a transformation.

## **Binding table**

View the columns of the binding table in graphical form, and click one or more columns to include in a transformation.

### Select All

Select all source columns and all destination columns. Use **Select All** when you want to create a many-to-many type of transformation mapping.

### **Delete All**

Delete all transformations. Select the mapping line representing the transformation you want to remove from the task before clicking **Delete**.

# See Also

Mapping Column Transformations

Data Driven Query Task

Multiphase Data Pump Functionality

# **Data Driven Query Task Properties (Queries Tab)**

Use this tab to query and set the properties of the source rowset.

# **Options**

## **Query Type**

Select the type of data driven query to execute. Options include Insert, Update, Delete, or User (stored procedure).

The query type labels are suggestions only. You can use a **Query Type** of **Insert** to perform an Update, a **Query Type** of **Delete** to perform an Insert, and so on. For readability, you may want to use the label that matches the type of query you plan to use.

## Build

Use Data Transformation Services (DTS) Query Designer to create the parameterized SQL statement to execute.

### **Parse/Show Parameters**

Check the syntax of the SQL statement and list any parameters mapped to binding columns on the **Transformations** tab.

### Destination

Indicate the binding column for a mapped parameter.

## Parameters

Indicate the corresponding parameter mapped to a binding column.

# See Also

Building a Data Driven Query

Data Driven Query Example: Changing Customer Accounts

# **Data Driven Query Task Properties (Lookups Tab)**

Use this tab to define a lookup query. Before defining a lookup query, you must first create connections for the source, binding, and lookup tables, and specify the source and binding tables. For more information, see <u>Lookup Queries</u>.

# **Options**

### Name

Type a name for the lookup.

## Connection

Select an existing connection on which to execute the lookup query.

## Cache

Specify the number of lookup results saved in a cache. Caching is especially useful with lookups if the number of rows being transformed is large and you are querying on a small number of rows in the lookup table.

# Query

Build the lookup query in Data Transformation Services (DTS) Query Designer. Alternatively, you can paste the query from a text editor into the SQL pane.

# Add

Create a new lookup query, which will be listed as a row in the **Lookups** table.

## Delete

Remove the highlighted lookup query from the **Lookups** table.

# See Also

Configuring a Simple Lookup Query

Using ActiveX Scripts in DTS

# **Data Driven Query Task Properties (Options Tab)**

Use this tab to send rows to an exception file, determine the format of that file, and set the error count at which package execution should cease. The exception files are stored on a local or mapped drive.

# **Options**

# **Exception file area**

## Name

Specify the path and name of the file where exception records will be written. If the file does not exist at package run time, the file will be created. The file does not have a default extension assigned to it.

Click the browse (...) button to search the local computer or mapped drives for an existing exception file. If an existing exception file is used, the status and error information for the package will be appended.

# File type area

# 7.0 format

Select to save the exception file in 7.0 format. This format is useful if an exception file parser was written for Microsoft® SQL Server<sup>TM</sup> version 7.0 exception files, as exception files using this backward-compatible format can still be used in the parser.

## **Error text**

Specify that any errors encountered during the task execution be recorded. Selecting this option records information such as the Data Transformation Services (DTS) package name, execution start and completion time, and other data in the exception log.

## Source error rows

Specify that a separate exception file be created to contain all the rows from

the source data that did not get written to the binding table. Formatting of the file is done according to the **File format properties** specified. The file name will be the same as the name specified in the **Name** field with the extension **.Source** appended to the name, and the file will be located in the same directory as the exception file.

### **Dest error rows**

Create a separate exception file to record error information on rows that fail when attempting to write to the destination. The file name will be the same as the name specified in the **Name** field with the extension **.Dest** appended to the name, and the file will be located in the same directory as the exception file.

# File format area

## **Row delimiter**

Select the delimiter used to separate rows of data in the exception file. A carriage return/line feed **{CR}{LF}** is used by default.

## Column delimiter

Select the delimiter used to separate the columns of data in the exception file. A vertical bar is used by default.

# **Text qualifier**

Specify which character marks are to be used in the delimited data file to qualify text. Choose from: **Double Quote {''}**; **Single Quote {'}**; and **<none>**. You also can type in a character to use as the text qualifier.

# Data movement area

## Max error count

Set a limit on the number of errors allowed before processing is terminated for the task. When the **Max error count** value is exceeded, task execution is terminated. The default is zero, which means that the task will terminate upon the first error.

# Fetch buffer size

Set the number of rows of data being fetched at the source during data movement. Generally, you should not need to adjust this value unless it is necessary to optimize the characteristics of the data provider.

## **First row**

Specify the first row of data to be moved. This is useful if the first row consists of column headings, or, if the first part of a data source has been copied, you can set this value to the row number where processing stopped in an earlier data pump operation.

## Last row

Specify the last row of data to move.

# See Also

Tasks That Transform Data

Data Driven Query Task
# **Date Time String Transformation Properties**

Use this dialog box to select the format for the source and destination date-time data used in a transformation. You can reformat the date and time source data and copy it to the destination using any of the available formats.

# **Options**

#### Source

Use these options to specify the format of the source data.

#### **Date format**

Select an available date-time format from the list.

#### Preview

View the current date and time in the selected date format.

#### Destination

Use these options to specify the format of the destination data.

#### **Date format**

Select an available date-time format from the list.

#### Preview

View the current date and time in the selected date format.

## Year 2000 cutoff date

Use to specify an integer that represents the cutoff year for interpreting twodigit years as four-digit years. A two-digit year that is less than or equal to the last two digits of the **Year 2000 cutoff date** is in the century that precedes the **Year 2000 cutoff date**. A two-digit year that is greater than the last two digits of the **Year 2000 cutoff date** is in the same century as the **Year 2000 cutoff date**.

## Naming

Rename date and time information or reset the language used when copying date and time information for a transformation.

# See Also

Date Time String Transformation Transformation Types Mapping Column Transformations

# **Define Columns**

Use this dialog box to specify the columns and their attributes when the destination is a text file.

# **Options**

#### **Table columns**

Define the columns in the text file. By default, all columns selected in the source are copied to the destination.

#### Name

Select whether or not to **<ignore>** a column in the destination.

## Туре

Select whether the data in the column is **quotable** or **not quotable**.

#### Size

Type the column size for a selected destination column.

## **Binary**

Select whether or not the data in a destination column is binary.

## **Populate from Source**

Copy the column schema as is from the source to the destination, overwriting any changes. You can still modify the columns and column attributes after selecting this option.

## Execute

Use the values entered in this dialog box to define the destination.

# **Define Row Width**

Use this dialog box to define the row width for data in a text file when the file does not contain delimiters (you must have selected **None** for **Row delimiter**). Select the red line and drag it to specify the number of characters you want in a row, shown in the **Value** box. Alternatively, you can click on the arrows alongside the **Value** box to set the row width.

# **DTS Package Properties (General Tab)**

Use this tab to view the general properties of a Data Transformation Services (DTS) package.

# **Options**

#### Name

View the package name.

## Description

Type a description of the text package.

## Package GUID

View the globally unique identifier (GUID) of the package, a 128-bit unique number.

## Version GUID

View the GUID of the package version, a 128-bit unique number. If only one version of the package exists, this number is the same as the package GUID; if more than one version exists, the version GUID is different.

## **Creator name**

View the name of the user who created the package (the DTS **Package.CreatorName** property). On computers running Microsoft® Windows® 98, this field may be blank; when it is not blank, it contains both the domain and user names.

## Computer

View the name of the computer on which the package was created (the Microsoft Win32® computer name, stored in the **Package.CreatorComputerName** property).

## Date

View the date on which the package was created (the **Package.CreationDate** 

property).

# **Priority class**

Specify the Windows process priority (the **Package.PackagePriorityClass** property). Possible values are **Low**, **Normal**, and **High**.

## Limit the maximum number of tasks executed in parallel to

Specify the maximum number of tasks that can execute concurrently. This can be any positive 32-bit integer. The default value is 4.

Setting this value too high can slow package execution; setting the value too low can slow simultaneous step execution.

See Also

Managing DTS Package Properties

# **DTS Package Properties (Global Variables Tab)**

Use this tab to view information about global variables, which can be referenced by any Microsoft® ActiveX® script in a Data Transformation Services (DTS) package. These scripts can be used to customize tasks, workflow steps, and transformations. Global variables defined on this tab remain in scope for the life of the package.

The values of global variables created in this tab are saved with the package. Global variables can also be created dynamically and used in ActiveX scripts; for example:

```
DTSGlobalVariables("NewGlobalVariable").Value = 5
```

Global variables created in this manner are not saved with the package and do not appear on the **Global Variables** tab.

# **Options**

## Name

Specify the global variable name. You can edit an existing name, or you can type a new name in the cell after clicking **New**.

# Туре

Specify the global variable data type from the list.

## Value

Specify the initial value assigned to the variable.

## **Explicit Global Variables**

Specify that global variables used in ActiveX scripts must be defined in this dialog box or in the **Execute SQL Task** dialog box before use. By setting **Explicit Global Variables**, you force yourself to declare every global variable, just as you do in Microsoft Visual Basic® when you turn on **Option Explicit**.

New

Add a global variable to the package. Clicking **New** inserts an empty row in the **Variables** table.

## Delete

Delete a selected global variable from the package.

# See Also

Using Global Variables with DTS Packages

# **DTS Package Properties (Logging Tab)**

Use this tab to save information regarding a Data Transformation Services (DTS) package execution. You can specify where you want to save the errors for a package, and decide whether the package should stop when an error is encountered. You can also have the package execution status saved to Microsoft® SQL Server<sup>™</sup> and specify to stop the package when problems are encountered with the log file.

# **Options**

#### Log package execution to SQL Server

Specify that you want to save the package and step execution data into the **msdb** database. Information about the package will be saved in the **sysdtspackagelog** file, while information about each step in the package will be saved in the **sysdtssteplog**. Steps that are not executed have no information stored for them in the log.

#### Server

Specify the name of the server whose **msdb** database will be the storage area for the logging information.

#### **Use Windows Authentication**

Specify to use Windows Authentication for login to the instance of SQL Server where the **msdb** database resides.

#### **Use SQL Server Authentication**

Specify to use SQL Server Authentication for login to the SQL Server database where the **msdb** database resides.

#### User name

Specify a user name for the **msdb** database connection.

#### Password

Specify a password for the **msdb** database connection.

## Fail package on log error

Specify that a failure to write the log to the server will stop the package from executing further. Log errors can be caused by having the event log buffer sized too small, or renaming the server log table.

#### **Delete Logs**

Specify that you want immediately to delete all logs of this package from the **msdb** database saved .

## Error file

Specify the name of the file where package and step status and error information will be written. This file will contain a list of the steps not executed, in addition to the steps that were executed and their result. The file can be on a local drive or on a mapped drive. If the file does not exist at package run time, the file will be created. The file does not have a default extension assigned to it; you must put the extension on the file name. The most common extension is .txt.

#### Browse

Display the **Select File** dialog box, where you can search the local computer or mapped drives for an existing error file. If an existing error file is used, the status and error information for the package will be appended.

#### Fail package on first error

Specify that a first step failure stops the entire package. Any remaining steps or tasks are not run. If this option is not selected, then the package will continue to run regardless of any failures in any of the steps. The package will complete with a successful status.

## Write completion status to event log

Specify that you want to write the package execution status to the Microsoft Windows® application log. This option is only available on computers running Windows NT 4.0 and Windows 2000. For more information, see <u>How to view the Windows application log (Windows)</u>.

See Also

Using DTS Package Logs

# **DTS Package Properties (Advanced Tab)**

Use this tab to specify data lineage properties, Microsoft® SQL Server<sup>™</sup> 2000 Meta Data Services scanning options, and transaction settings for Data Transformation Services (DTS). These allow you to do the following:

- The data lineage feature of DTS allows you to determine the source of any piece of data and the transformations applied to that data.
- Meta Data Services scanning options are settings for relating objects referenced by the package to catalog meta data in Meta Data Services.
- Transaction capabilities allow you to assign tasks in a workflow to a transaction, and commit and roll back individual steps based on the success or failure of the transactional unit. The settings on the **Advanced** tab allow you to turn on transaction capabilities, and set their commit mode and isolation level.

**Note** Not all OLE DB providers support transactions or specific transaction capabilities. For more information, see the documentation for the individual providers.

The first two groups of these options can only be used if you are saving a DTS package to Meta Data Services. If you attempt to save a package to any other location or format after you have clicked any of these options, DTS Designer prompts you to save the package to Meta Data Services.

# **Options**

#### Name

View the package name.

## Show lineage variables as source columns

Add global data lineage variables to the package, but do not write them to Meta Data Services. Clicking this option by itself is useful if you want to create a custom task to write lineage tracking and auditing information.

#### Write lineage to repository

Specify to always write data lineage variables to the repository database when saving a package.

## Options

Set Meta Data Services scanning options for the package.

#### **Use transactions**

Allow the definition and use of transactional units of work in the package.

#### Commit on successful package completion

Specify that each individual SQL statement is a transaction. If the statement completes successfully, the transaction is automatically committed; if the statement has an error, the statement is rolled back (the default setting for OLE DB).

When this setting is cleared, the DTS connections operate in implicit transaction mode. (The first SQL statement begins a transaction that remains in effect until DTS commits it or rolls it back. A new transaction is started by the next SQL statement executed after any commit or rollback.)

## **Transaction isolation level**

Select from one of these levels:

- Chaos. You can see uncommitted changes made by other transactions, but update locks are not held to the end of the transaction. Rollback is not supported. This isolation level is not supported by SQL Server.
- Read Committed. You cannot see changes made by other transactions until those transactions are committed.
- Read Uncommitted. You can see uncommitted changes made by other transactions.

- Repeatable Read. You are guaranteed not to see any changes made by other transactions in values it has already read.
- Serializable. This option guarantees that all concurrent transactions will interact only in ways that produce the same effect as if each transaction were executed entirely one after the other.

#### Use OLE DB service components

Select to instantiate the OLE DB provider data source objects using the OLE DB service components (**IDataInitialize::CreateDBInstance**,the default), or clear to instantiate the data source objects directly with **CoCreateInstance**.

OLE DB service components provide services like session pooling and **IRowsetChange**, which may not be supported by some OLE DB providers. This setting is ignored by the DTS providers (PackageDSO, RowQueue, FlatFile) and by the Microsoft OLE DB Provider for SQL Server.

# See Also

Incorporating Transactions in a DTS Package

Recording Data Lineage in DTS

Managing DTS Package Properties

# **Data Transformation Services Query Designer**

Use the simple graphical interface of the Data Transformation Services (DTS) Query Designer for building SQL commands.

The DTS Query Designer contains a selection box that lets you choose the server to connect to, and several synchronized panes that let you graphically manipulate data and enter SQL command text to build queries.

# **Options**

#### Linked server

Select the server on which the connected database you want is located. Any servers you connected to in Data Transformation Services (DTS) are available, including any defined linked servers.

#### **DTS Query Designer Panes**

The DTS Query Designer contains the **Table/View List pane**, the **Diagram pane**, the **Grid pane**, the **SQL pane**, and the **Results pane**:

#### Table/View List pane

This pane displays a list of tables or views in the connected database. When you change the server, the list of tables and views refreshes. To use this pane to build queries, select a table or view and drag it onto the Diagram pane.

#### **Diagram pane**

This pane displays the tables and other table-structured objects that you are querying. Each rectangle represents a table or table-structured object and shows the available data columns as well as icons that indicate how each column is used in the query. Joins are indicated by lines between the rectangles.

#### Grid pane

This pane contains a spreadsheet-like grid in which you specify options, such as which data columns to display, what rows to select, how to group

rows, and so on. For more information, see Grid Pane.

### SQL pane

This pane displays the SQL statement for the query or view. You can edit the SQL statement created by the DTS Query Designer or you can enter your own SQL statement. It is particularly useful for typing SQL statements that cannot be created using the **Diagram** and **Grid** panes, such as Union queries. For more information, see <u>SQL Pane</u>.

## **Results pane**

This pane shows a grid with data retrieved by the query or view. In the DTS Query Designer, the pane shows the results of the most recently executed Select query. You can modify the database by editing values in the cells of the grid, and you can add or delete rows. For more information, see <u>Results Pane</u>. In the View Designer, the results pane shows the contents of the view.

You can create a query or view by working in any of the panes: you can specify a column to display by choosing it in the **Diagram** pane, entering it into the **Grid** pane, or making it part of the SQL statement in the SQL pane. The **Diagram**, **Grid**, and **SQL** panes are synchronized. When you make a change in one pane, the other panes automatically reflect the change.

# See Also

Diagram Pane Grid Pane Navigating in the Query Designer Results Pane SQL Pane

# **Dynamic Properties Task Properties**

Use this tab to view the package properties that will be assigned at Data Transformation Services (DTS) package run time.

#### Description

Type a description of the Dynamic Properties task. This description appears on the DTS Designer design sheet.

#### **Destination Property**

View the name of the package property whose value will be assigned at run time.

#### Source Type

Select one of the following external sources from which a property value is assigned:

An .ini file

A data file

A query

A global variable

An environment variable

A constant

#### **Source Value**

View the current value of the property, which may change each time the package is run.

#### Add

Display the **Package Properties** dialog box, where you can select a new package property to assign a value dynamically.

#### Edit

Click to change the dynamic assignment of a selected property in the

# Change list.

# Delete

Click to permanently remove a dynamic property assignment from the **Change** list.

# See Also

Dynamic Properties Task

# **Dynamic Properties Task: Package Properties**

Use this dialog box to view and select a property to be set dynamically.

The left pane displays a tree view of Data Transformation Services (DTS) package properties. Expand the appropriate nodes to navigate to the group containing the property whose value you want assigned dynamically.

The right pane lists the names and default values of the properties belonging to the property group selected in the left pane.

# Options

# Add

After selecting a property from the right pane, click to display the **Add/Edit Assignment** dialog box, which you use to make the property assignment.

# Close

Return to the **Dynamic Properties Task Properties** dialog box.

# Leave this dialog box open after adding a setting

Make multiple property assignments at a time. If this box is cleared, the **Add/Edit Assignment** dialog box is closed after a setting is added, and the **Dynamic Properties Task Properties** dialog box is displayed.

# See Also

**Dynamic Properties Task** 

# **Edit All Package Properties**

Use this dialog box to select a Data Transformation Services (DTS) package property whose value you want to view or modify. You cannot modify all package properties.

The **Edit All Package Properties** dialog box is divided into left and right panes:

- The left pane contains an expandable tree view of all the property groups in the current package. To expose the properties, click on the plus icons to expand branches in the tree view. After exposing the property group containing the property you want to view or modify, select the property by clicking on its name.
- The right pane contains a list of properties for the node selected in the left pane. Use it to select the individual property whose value you want to view or modify.

# **Options**

## Edit

View or edit a property value after selecting a property in the right pane.

#### Close

Save changes and close the dialog box.

# See Also

Editing DTS Package Properties with Disconnected Edit

# **Edit Property**

Use this dialog box to view or change a selected property. If a property cannot be edited, you can view the data, but the **Type** and **Value** boxes are disabled.

# **Options**

#### Name

View the name of the property selected in the **Edit All Package Properties** dialog box.

## Туре

Select a data type for the property from the list.

**WARNING** Not all data types may be valid for the selected property, and Data Transformation Services (DTS) Designer does not perform any validation checking. Make certain any data type changes you make are valid or you may disable the DTS package.

## Value

View or change a value for the property.

# See Also

Editing DTS Package Properties with Disconnected Edit

# **Execute Package Task Properties (General Tab)**

Use the Execute Package task to specify another Data Transformation Services (DTS) package to run as part of package execution, and to pass information contained in parent package global variables to the child package.

# **Options**

## Description

Type a text label for the Execute Package task, which will appear on the Data Transformation Services (DTS) Designer design sheet.

#### Location

Specify how the package that will be run by the Execute Package task was saved:

To Microsoft<sup>®</sup> SQL Server<sup>TM</sup>

To SQL Server 2000 Meta Data Services

To a structured storage file

#### Package name

Type the name of the package to be run by the task, or browse for the package.

**Note** If you select a specific version of the package to be run in the **Select Package** dialog box, you will always execute that version of the package. If you select a package node in the **Select Package** dialog box, the latest version of the selected package is always run.

## Password

Type the password of the package to be run by the task, if one exists.

## Package ID

View the globally unique identifier (GUID) associated with the package to be run.

## Server

Select the name of the server containing the external package to be run from the list of available servers (SQL Server and SQL Server Meta Data Services only).

## **Use Windows Authentication**

Specify Windows Authentication for login (SQL Server and Meta Data Services only).

#### **Use SQL Server Authentication**

Specify SQL Server Authentication for login (SQL Server and Meta Data Services only).

#### User name

Specify a user name for a new connection (SQL Server and Meta Data Services only).

#### Password

Specify a password for a new connection (SQL Server and Meta Data Services only).

# See Also

**Execute Package Task**
# **Execute Package Task Properties (Child Package Globals Tab)**

Use this tab to specify global variable information that will be passed to the Data Transformation Services (DTS) package to be executed. Make sure that the child package contains definitions for the global variables specified here or the Execute Package task will fail.

Information about global variables is displayed in each row of the **Variables** table. The buttons at the bottom of the table control the addition or deletion of global variables. You can select a cell in the table to enter or change information.

### Options

#### Name

Specify the global variable name. You can edit an existing name, or type a new name in the cell after clicking **New**.

#### Туре

Specify the global variable data type from the list.

#### Value

Specify the initial value assigned to the variable.

#### New

Add a global variable to the package. Click **New** to insert an empty row in the **Variables** table.

#### Delete

Delete a selected global variable from the package.

#### See Also

Execute Package Task

Using Global Variables with DTS Packages

### **Execute Package Task (Parent Package Globals Tab)**

Use this tab to select those global variables in the parent package that will pass information to the child package. You can use existing global variables or create new ones. Global variables defined in this dialog box are created and executed at run-time. You can also define global variables in the **DTS Package Properties** dialog box.

#### **Options**

#### Name

Specify the global variable name. You can edit an existing name, or type a new name in the cell after clicking **New**.

#### New

Add a global variable to the package. Click **New** to insert an empty row in the **Variables** table.

#### Delete

Delete a selected global variable from the parent package global variables list.

#### See Also

Execute Package Task

Using Global Variables with DTS Packages

### **Execute Process Task Properties**

Use this tab to specify the properties to run an executable program or batch file as a task.

### **Options**

#### Description

Type a description of the Execute Process task. This description becomes the label on the Data Transformation Services (DTS) Designer design sheet.

#### Win32 process

Specify the name of the executable file (.exe) or batch file (.bat) that will be run.

#### Browse

Display the **Select File** dialog box, where you can search the local computer and mapped drives for the executable or batch file.

#### Parameters

Specify any command prompt parameters required for the executable file or batch file. Separate multiple parameters with a space.

#### **Return code**

Specify the return code you want upon successful execution of the process.

#### Timeout

Specify the number of elapsed seconds within which the process can successfully execute. A value of 0 means that no time-out value is used, and the process runs until completion or error.

#### **Terminate process after timeout**

Terminate the process after the time-out period has been reached.

#### See Also

Execute Process Task CreateProcessTask Object CreateProcessTask2 Object

### **Execute SQL Task Properties**

Use this dialog box to name the task, specify the data connection, specify the SQL statements to execute, and set the connection time-out.

#### **Options**

#### Description

Type a description of the Execute SQL task. This description becomes the label on the Data Transformation Services (DTS) Designer design sheet.

#### **Existing connection**

Choose the connection to the appropriate data against which you want the SQL statement to run.

#### **Command Timeout**

Specify the value that determines the maximum number of seconds the execution can take before timing out. A value of zero indicates an infinite amount of time.

#### **SQL Statement**

Specify the SQL statement to execute. You can separate multiple queries with the GO statement.

#### **Parameters**

Display the **Parameter Mapping** dialog box, where you can map global variables as either input parameters to your SQL statement or as output parameters to contain the results of a query. If you create an SQL statement with a parameter marker and the **Parameters** button is not enabled, the selected provider does not support parameterized queries.

#### **Parse Query**

Check the syntax of the SQL statements. The OLE DB provider provides syntax checking for standard SQL syntax. The syntax checking will not catch errors in complicated expressions, nor does it support the use of non-standard SQL constructs, such as parameter markers and comments. Error messages related to incorrect syntax should be verified and can be ignored if you find that the SQL statement is valid.

#### **Build Query**

Create the SQL statement to execute using DTS Query Designer, a graphical query-building tool.

#### Browse

Display the **Selected File** dialog box, where you can select a file containing prewritten SQL statements. When a file is selected, the SQL contained in that file is copied into the SQL statement text area.

#### See Also

Execute SQL Task

ExecuteSQLTask Object

## **Executing DTS Package**

Use this dialog box to view the execution progress of the Data Transformation Services (DTS) package overall, as well as the execution status of the individual tasks composing the package.

To display error information about a task, double-click the task that did not execute correctly.

#### **Options**

#### Step

View the name of the task that executes in the step.

#### Status

View the progress of the executing step, and whether the step completed successfully or failed. For tasks that report the number of rows processed (shown in parentheses), the number indicates the total number of rows read from the source, not the number of rows that were successfully processed.

### See Also

Executing a DTS Package

### File Transfer Protocol Task Properties (Location Tab)

Use this tab to specify and configure a remote server or Internet location from which to download data.

#### **Options**

#### Description

Type a text label for the File Transfer Protocol (FTP) task.

#### Source

Select whether to download data from a **Directory** location in your network or from an **Internet Site**. Different options become available depending on which selection you choose.

#### **Directory path**

If you choose to download data from a **Directory** location, enter the directory path. Click the browse (...) button to search for the directory from which to download data.

#### **FTP Site**

If you choose to download data from an **Internet site** location, enter the FTP address for the source data.

#### Username

Type the user name required for access to the FTP site.

#### Password

Type the password required for access to the FTP site.

#### Number of retries

Specify the attempted number of connections if the first one fails.

#### Destination

Specify information about the destination for the FTP data.

**Directory path** Type the FTP site location for the destination files.

### See Also

File Transfer Protocol Task

### **File Transfer Protocol Task Properties (Files Tab)**

Use this tab to select which files to copy between File Transfer Protocol (FTP) sites.

### **Options**

#### Source

View information about the source files and select the files to be copied. Double-clicking a file moves it to the **Destination** list.

#### Destination

View information about the destination files, and select the files that you do not want copied. Double-clicking a file removes it from the **Destination** list.

>

Move a selected file from the **Source** list to the **Destination** list.

>>

Move all files in the **Source** list to the **Destination** list.

#### <

Remove a selected file from the **Destination** list.

#### <<

Remove all files from the **Destination** list.

#### Refresh

Redisplay the listing of files, in case files have been added to or removed from the specified location.

#### Overwrite

Select whether you want to overwrite existing files with the same names or create new files when the copies are made.

See Also

File Transfer Protocol Task

### Font

Use this dialog box to format the Transact-SQL code in the query box.

### Options

#### Color

Specify a Transact-SQL content element to assign color coding and formatting.

#### Foreground

Assign a color for the selected content element.

#### Background

Assign a background color for the selected query element.

#### Font

Assign a font to the selected Transact-SQL content element.

#### Size

Assign a font size to the selected Transact-SQL content element.

#### Sample

View an example of the selected Transact-SQL content element with the assigned font attributes.

#### **Reset All**

Reset all options to their original default values.

## **Global Variables**

Use this dialog box to view or change information about global variables in the Data Transformation Services (DTS) package. Global variables defined in this dialog box are saved with the package, and they are usable from any transformation scripts written at the task, step, and column-mapping levels. You can also define global variables in the **DTS Package Properties** dialog box.

Information about global variables is displayed in each row of the **Variables** table. The buttons at the bottom of the table control the addition or deletion of global variables. You can select a cell in the table to enter or change information.

### Options

#### Name

Specify the global variable name. You can edit an existing name, or type a new name in the cell after clicking **New**.

#### Туре

Specify the global variable data type from the list.

#### Value

Specify the initial value assigned to the variable.

#### New

Add a global variable to the package. Click **New** to insert an empty row in the **Variables** table.

#### Delete

Delete a selected global variable from the package.

#### See Also

Using Global Variables with DTS Packages

### **Message Queue Message Properties**

Use this dialog box to define the type and content of messages that will be sent to a message queue.

### **Options**

#### Message type

Select one of the following message types from the list:

String Message, which contains a text string you specify.

**Data File Message**, which contains a Data Transformation Services (DTS) package ID, a version ID, and the name and contents of a data file.

**Global Variables Message**, which contains a DTS package ID, a version ID, and the name, type, and value of one or more variables.

#### Message

Choose whether you want the message queue task to **Send messages** to other packages or **Receive messages** from other packages.

#### String message

Type the text of the message if you selected a **Message type** of **String Message**.

#### File name

Type the path and name of the data file containing the package information.

#### **Global variables**

Displays a list of the package global variables that will be sent to the message queue.

#### Name

After clicking **New**, select a global variable to include in the message from the list of available package global variables.

#### Туре

View the data type of the selected package global variable.

#### Value

View the initial value of the selected package global variable.

#### **Create Global Variables**

Create new package global variables that you can include in the message queue task.

#### New

Create a global variable entry to include in the message.

#### Delete

Delete a global variable entry. Deleting a global variable entry does not delete the global variable definition from the package.

#### **Global variables**

Display a list of the global variables that will be sent to the message queue.

#### See Also

Message Queue Task

Message Types

### **Message Queue Task Properties**

Use this dialog box to configure a Message Queue task, which allows you to use Message Queuing to send and receive messages between Data Transformation Services (DTS) packages.

#### **Options**

#### Description

Type a text name for the message queue task.

#### Message

Select whether you want the message queue task to **Send messages** or **Receive messages**.

#### Queue

Specify the queue from which you will send or receive messages. The format for the queue is: *computer\_name\queue\_type\$\queue name*, where *queue\_type* can be Outgoing, Public, Private, or System.

#### **Send Message Options**

If you select **Send messages** in the **Message** box, the following options are available:

#### Messages to be sent

Displays information on each message defined for this message queue task:

#### Number

View the automatically generated number that designates the order in which a message was created.

#### Message type

View whether the message consists of a data file, global variable, or string.

#### Add

Define a new message to add to the message queue task.

#### Edit

Edit an existing message within the message queue task.

#### **Receive Message Options**

If you select **Receive messages** in the **Message** box, different options become available depending on the **Message type** you select:

#### Message type

Select one of the following message types from the list:

String Message, which contains a text string you specify.

**Data File Message**, which contains a DTS package ID, a version ID, and the name and contents of a data file.

**Global Variables Message**, which contains a DTS package ID, a version ID, and the name, type, and value of one or more variables.

#### Only receive message from a specific package or version

Lets you filter the possible messages the message queue task can receive:

**No filter**. Select this option to turn message filtering off for the task.

**From package**. Select this option to specify a package to use as a filter. Only messages from versions of the specified package will be received.

**From version**. Select this option to specify a package version to use as a filter. Only messages from a package with the specific version identifier will be received.

**Identifier**. View the globally unique identifier (GUID) of the package or package version to be used as a filter.

... Browse for a package or version to use as a filter for receiving messages.

#### Save file as

Save the contents of a data file message into the specified file, by typing the file path and name or by browsing for a file in which to save the contents of a data file message.

#### Compare

Specify a filter to use when receiving a string message. The string message is not received by the Message Queue task unless it matches the filter. Use one of the following:

None, to specify no filter.

**Exact match**, to specify that the message content must exactly match the string entered in the **Compare string** box.

**Ignore case**, to specify that the message content must match that of the string entered in the **Compare string** box, irrespective of case.

**Containing**, to specify that the message content must contain the string entered in the **Compare string** box.

#### **Compare string**

Type the string to be used as a filter when evaluating the string message.

#### Overwrite

Select to overwrite the data in an existing file when saving the contents of a data file message.

#### Time out after

Select to specify a time-out period in which to receive the message.

#### See Also

Message Queue Task

Message Types

### **Middle of String Transformation Properties**

Use this dialog box to copy a substring from the source column, transform it, and copy the result to the destination column. You specify the substring by providing a start position and a maximum number of characters to include. Other options are available that allow you to trim white space from the string and change case.

#### **Options**

#### **Substring Options**

Use these options to locate the substring to be copied.

#### Start position (1 based)

Define the starting character position (the index starts at the number 1) from which the substring will be copied.

#### Limit number of characters to

Specify the number of characters to be copied. By default, all characters in the string, including the character in the start position, will be copied (default value of 0).

#### **Trimming Options**

Select from which location in the string to remove white space.

#### Trim leading white space

Remove any white space preceding the first character of the string data.

#### Trim trailing white space

Remove any white space after the last character of the string data.

#### Trim embedded white space

Remove any white space in between the first and last characters of the string data.

#### **Case Options**

Select whether to change the case of the string data as part of the

transformation.

#### Do not change case

Leave the case of the string data intact after performing any trim operations.

#### Uppercase

Change the case of the string data to uppercase after performing any trim operations.

#### Lowercase

Change the case of the string data to lowercase after performing any trim operations.

### See Also

Middle of String Transformation

Trim String Transformation

### **Parameter Mapping (Input Parameters Tab)**

Use this tab to specify the global variables to be assigned to the parameters in the SQL statement. The parameters, indicated by question marks in the SQL statement, are parsed left to right and are represented in the **Parameters** column as **Parameter1**, **Parameter2**, and so on.

#### **Options**

#### **Input Global Variables**

Select the global variable whose value you would like to use to replace the parameter marker at run time.

#### Parameters

View a read-only list of parameter markers from the SQL statement.

#### **Create Global Variables**

Display the **Global Variables** dialog box, where you can create new Data Transformation Services (DTS) package global variables.

#### See Also

Using Parameterized Queries in DTS

Execute SQL Task

Using Global Variables with DTS Packages
# **Parameter Mapping (Output Parameters Tab)**

Use this tab to specify the global variables that will contain the data returned from the query.

# **Options**

#### **Output Parameter Type Area**

#### None

Specify that no data be returned from the query, or, if data is being returned, that no data be stored in a global variable.

#### **Row Value**

Specify that each column returned from the query be stored in a separate global variable. The grid will populate with the corresponding number of columns in the results set. The columns will be listed in order as **Parameter1, Parameter2**, and so on. Automatic mapping to existing global variables is done ordinally. If you do not want to save the value of a column being returned, type **<none>** in the **Output Global Variable** field. To change which column is being stored in which global variable, use the list in the **Output Global Variable** cell to select the global variable that should hold the data. If the global variable you want to use does not exist, click **Create Global Variable**. After the global variable has been created, it is now ready to be used in the parameter mapping.

#### Rowset

Specify that you want to store the data returned from the query in a single global variable. The global variable will hold the entire rowset. This global variable can be used in subsequent tasks as a disconnected Microsoft® ActiveX® Data Objects (ADO) recordset. If no data is returned from the query, then the global variable will contain nothing. If the global variable you want to use does not exist, click **Create Global Variable**. After the global variable has been created, it is now ready to be used in the parameter mapping.

#### **Rowset List**

Specify the name of the global variable that will hold the rowset.

#### **Parameter Mapping Area**

#### **Parameters**

Display a read-only list of the columns being returned from the SQL statement, if applicable.

#### **Output Global Variable**

Specify the global variable to contain the data in the column when the data is being returned as a row value.

#### **Create Global Variable**

Display the **Global Variables** dialog box, where you can type a new global variable for the Data Transformation Services (DTS) package.

## See Also

Using Parameterized Queries in DTS

Execute SQL Task

Using Global Variables with DTS Packages

# **Read File Transformation Properties**

Use this dialog box to copy the entire content of a file specified by a source column to a destination column.

# Options

#### Directory

Type the directory location for the files from which you want to read data. At run time, each of the entries in the source column is appended to this directory name to form a read path. If a file exists at that location, it is opened and the contents (possibly translated) are copied to the destination column. The directory name may start with a disk drive or a Universal Naming Convention (UNC).

You also can browse for the location of the directory containing the files from which you want to read data.

#### File type

Select whether the files that the transformation reads data from are **ANSI**, **Unicode**, or **OEM**.

#### Error if file not found

Select to stop processing and fail the read file transformation if a file specified in the source column is not found in the directory location.

## See Also

**Read File Transformation** 

# **Register Custom Task**

Use this dialog box to register a custom task. Custom tasks are written in languages such as Microsoft® Visual Basic® or Microsoft Visual C++®. After you create a custom task, you can register the task and include it in the Data Transformation Services (DTS) Designer user interface. Then you can use the custom task when you create packages with DTS Designer.

# **Options**

#### **Task description**

Type a description of the custom task.

#### **Task location**

Type the path and file name of the dynamic-link library (.dll file) for the custom task, or browse for the location of that library.

#### **Icon location**

Type the path and file name of the icon for the custom task, or browse for the location of that icon.

#### Refresh

Clear data from the boxes.

#### Select icon

Select an icon for the custom task.

# See Also

DTS Custom Task

Building a DTS Custom Task

# Save DTS Package

Use this dialog box to save a Data Transformation Services (DTS) package to the Microsoft® SQL Server<sup>™</sup> **msdb** database, SQL Server 2000 Meta Data Services, a COM-structured storage file, or a Microsoft Visual Basic® file. You can also set package authentication information and passwords in this dialog box.

# **Options**

When you are saving a package, the following options are available:

#### Package name

Specify a unique name for the package. The **msdb** tables use this name as a primary key.

#### **Owner password**

Specify a password for the package to protect any sensitive user name and server password information in the package from unauthorized users. If the package has an owner password, the data is encrypted with the standard encryption API. This option is available only for packages saved to SQL Server or as a structured storage file.

#### User password

Set a password for a package user. This password allows a user to execute a package. However, this option does not allow a user to view the package definition. If you set the user password, you also must set the owner password. This option is available only for packages saved to SQL Server or as a structured storage file.

#### Location

Specify the format and location of the saved package. You can save to a SQL Server **msdb** database, to Meta Data Services, to a structured storage file, or to a Visual Basic file.

When you save the package to SQL Server or to Meta Data Services, you have

the following options:

#### Server name

Specify the name of the SQL Server installation storing the package.

#### **Use Windows Authentication**

Specify the security mode used to connect to SQL Server. The Windows Authentication used will be the Microsoft Windows® login of the user creating the package.

#### **Use SQL Server Authentication**

Specify the security mode used to connect to an instance of SQL Server.

#### User name

Specify a user name for the connection to an instance of SQL Server.

#### Password

Specify a password for the connection to an instance of SQL Server.

When you save the package to Meta Data Services, you have the following additional option:

## Scanning

Display the **Scanning Options** dialog box, where you specify how objects referenced by the package should be scanned into Meta Data Services. This capability allows you to relate source and destination objects in a package to database meta data (for example, primary and foreign keys in a table, indexes, and column information, such as data type) stored in Meta Data Services.

When you save the package as a COM-structured storage file or a Visual Basic file, you have the following options:

## File name

Specify the package file name and path. If the package is a structured storage file, it should be stored with the extension .dts. If the package is a Visual Basic file, it should be stored with the extension .bas.

# Browse (...)

Display the **Save As** dialog box, where you can specify the file name, extension, and storage location.

# See Also

Saving a DTS Package

# **Scanning Options**

Use the OLE DB scanner for Microsoft® SQL Server<sup>™</sup> 2000 Meta Data Services to import database schema information from an OLE DB data source and populate instances of the Database Information Model (DBM) in Meta Data Services. The scanner is passed an OLE DB provider, examines the schema, and creates a set of corresponding instance objects in Meta Data Services using the DBM and Data Transformation Services (DTS) Meta Data Services information models.

If you save a DTS package to Meta Data Services, you can set how the objects referenced by the package are related to SQL Server catalog meta data scanned into Meta Data Services. Meta data in this context refers to information such as:

- Primary and foreign keys.
- Column type, size, precision, scale, and nullability.
- Indexes.

# **Options**

#### Resolve package references to scanned catalog meta data

Enable the scanning options, which link the package to Meta Data Services meta data.

#### Use scanned catalogs if already present in Meta Data Services

Relate the package to Meta Data Services meta data if the meta data has already been saved (that is, by importing the meta data, or by a previous scan). This option is useful if the database schema has not changed, Meta Data Services meta data already exists, and saving time is a consideration. If column or meta data information has changed and this option is used, references of the package to Meta Data Services meta data may not be meaningful.

#### Scan all referenced catalogs into Meta Data Services

Scan everything about a database (all table, column, and meta data information) into Meta Data Services, even if the information is not used. This is safer than the previous option, but more costly in terms of time and performance.

#### Scan catalog if not already present in Meta Data Services

If a database is not present in Meta Data Services, add it. (This is the default selection if scanning options are enabled.)

#### Scan catalog always

Scan a database into Meta Data Services even if it is present already.

# See Also

Importing and Saving Meta Data in DTS

**Recording Data Lineage in DTS** 

# **Select Objects**

Use this dialog box to select the objects that you want to transfer to the new database.

**Note** Selecting a check box does not automatically mark the corresponding object to be transferred. The objects to appear are in the **Objects** list, from which you then select the objects to transfer.

# **Options**

#### Show all tables

Display all tables in the database in the **Objects** table.

#### Show all views

Display all views in the database in the **Objects** table.

#### Show all stored procedures

Display all stored procedures in the database in the **Objects** table.

## Show all defaults

Display all defaults in the database in the **Objects** table.

## Show all rules

Display all rules in the database in the **Objects** table.

## Show user-defined data types

Display all user-defined data types in the database in the **Objects** table.

## Select All

Select all objects in the database as eligible for transfer to the new database.

## Check

Place a check mark next to all selected objects.

## Uncheck

Remove the check mark from all selected objects.

# See Also

Copy SQL Server Objects Task

# Select Package

Use this dialog box to select from Data Transformation Services (DTS) package versions saved to a structured storage file or to Microsoft® SQL Server<sup>™</sup>.

# Selecting a Package Version Saved to a Structured Storage File

When you open a package saved to a structured storage file, if more than one package or package version is contained in the file, the **Select Package** dialog box appears. The dialog box contains a tree view of a package version history.

- At the top node, a text file icon and accompanying label indicates the storage location for the package.
- Package nodes are shown beneath the top node. Each of these nodes represents a different package. There can be multiple package nodes listed under a file, if different packages were saved under the same file name.
- Version nodes with dates, indicating the complete version history of a package, are displayed beneath a package node.

Double-clicking a package icon loads the latest version of that package version; double-clicking a version icon loads that specific package version.

# Selecting a Package Version Saved to SQL Server

The **Select Package** dialog box appears when you attempt to load a package saved to SQL Server with multiple versions. As with structured storage files with multiple versions, a tree view of the package history is displayed for selection. However, only versions for a single package are displayed; you do not save multiple packages to a single file, as is the case with a structured storage file.

Expand a package icon to view its version history, and then double-click on the package icon to load the latest version of the package, or click on a specific version to load that version.

# See Also

Saving a DTS Package to a Structured Storage File Saving a DTS Package to SQL Server DTS Package Execution Utilities

# **Send Mail Task Properties**

Use this tab to specify the properties that will direct the message to the appropriate recipient. The Send Mail task allows attachments to be sent with the message.

# **Options**

#### Description

Type a description of the Send Mail task. This description becomes the label on the Data Transformation Services (DTS) Designer design sheet.

#### **Profile Name**

Specify the appropriate e-mail profile name by selecting from the **Profile Name** list, which contains all locally-defined MAPI profiles.

#### Password

Type the profile password.

#### То

Specify the name of the e-mail recipient. Click the browse (...) button to display the **Address Book on <profile**> dialog box, where you select the address book that contains the name you want.

#### CC

Specify the names of any recipients who should receive carbon copies of this e-mail. Click the browse (...) button to display the **Address Book on** <**profile**> dialog box, where you select the address book that contains the name you want. From the list of names, select the recipient to whom you want the carbon copy to go.

#### Subject

Specify a subject for the e-mail.

#### Message

Type the e-mail message.

## Attachments

Specify the files to attach to the e-mail.

# Add

Display the **Select File** dialog box, where you can select the file that you want to attach to the e-mail.

# See Also

Send Mail Task

# **Text File Properties**

Use this dialog box to set the data format of the source or destination text file. There are two data formats from which to choose: delimited and fixed field.

# **Options**

#### Delimited

Specify that the source or destination text file uses delimiter characters to indicate columns.

#### **Fixed field**

Specify that the data in the source or destination text file is aligned into equal width columns.

#### **Skip rows**

Specify the number of rows you want skipped in a text file. This option only applies when the text file is used for source data.

#### First row has column names

Indicate that the first row in the specified file consists of column names.

#### File type

Specify whether the data file is ANSI (default), Unicode, or OEM.

#### **Row delimiter**

Specify whether the character that delimits rows in the data file is carriage return/line feed **{CR}{LF}** (default), carriage return **{CR}**, or line feed **{LF}**. You can also type another character.

#### Text qualifier

Specify whether quotation marks are to be used to qualify text in the data file.

# **Task References**

Use this dialog box to change the source or destination connections used in an already mapped transformation.

#### Task Name

Displays all the tasks associated with the source or destination that were modified.

#### **Clear Transformations**

Map transformations between source and destination fields, and set a default Copy Column transformation for each pairing; or clear for the previously defined transformations to remain intact.

#### Cancel

Keep the previously defined transformations intact.

**IMPORTANT** The application does not check that the fields are valid for the new source or destination chosen. It simply applies the selected action to the fields that were in existence at the time the transformations were made originally.

# See Also

Mapping Column Transformations

**Transformation Types** 

Copy Column Transformation

# **Testing Transformation**

Use this dialog box to view the execution progress of the Microsoft® ActiveX® script transformation you selected.

The transformation is run on the source data, and the results are copied to a temporary text file. The destination table is not affected. Because test mode sends data to a file rather than the actual destination, problems with the provider, the destination, or the data type overflow may not be detected.

# **Options**

#### **Execution Progress**

View the progress of the transformation being tested.

#### **Execution status**

View error information about a transformation that did not execute correctly. Double-click the step.

#### Done

Display the **ActiveX Script Transformation Properties** dialog box, where you can change the ActiveX script.

#### **View Results**

Display the **View Data** dialog box, where you can view the contents of the temporary file containing the results of the test transformation.

## See Also

Using ActiveX Scripts in DTS

Tasks That Transform Data

Transform Data Task

# **Transform Data Task Properties (Source Tab)**

Use this tab to set the properties of and query source data.

# **Options**

#### Description

Type a description of this Transform Data task.

#### **Table/View**

Select a table or a view from the data source specified on the connection. There is no guaranteed ordering of rows.

#### SQL query

Specify that an SQL statement retrieve the data from the data source. Type the SQL statement.

#### Parameters

Map global variables to input parameters by specifying an SQL query that includes parameters (using one or more question marks as a parameter placeholder).

#### Preview

View the data in the selected source table.

**Note** If your source connection is to a dBase data source, and you specify an SQL query, the source data may not be available for viewing.

## **Build query**

Use the Data Transformation Services (DTS) Query Designer, a visual query building tool, to create the SQL statement to execute.

#### Browse

Specify the location of an SQL query to execute.

#### **Parse query**

Check the syntax of the SQL statement prior to execution.

# See Also

Transform Data Task

**DTS** Connections

Using Parameterized Queries in DTS

# **Transform Data Task Properties (Destination Tab)**

Use this tab to configure the destination table or storage location for the results of the task. The properties to be configured will vary, depending on the characteristics of the destination data source.

# **Options**

#### Table name

Select the data destination of the transformation from the list.

#### Create

Specify that a new table should be created as the destination. A destination table or storage area must exist before you can specify transformations.

#### Name

View the names of the destination columns.

#### Туре

View the destination native data type using OLE DB data type mappings.

#### Nullability

View whether the destination columns allow null values. The default is true.

#### Size

View the width of destination columns where applicable.

#### Precision

View the precision of destination columns where applicable.

#### Scale

View the number of digits in scale where applicable.

## See Also

Transform Data Task
DTS Connections

# Transform Data Task Properties (Transformations Tab)

Use this tab to select and graphically map the source and destination columns composing each transformation. Graphically mapping a transformation is not required. You can also map transformations using selection boxes in the **Transformation Options** dialog box.

Column mappings are represented on this tab by arrows connecting columns in the **Source** and **Destination** boxes. By default, the Transform Data task maps all columns using one-to-one column mappings.

# Options

#### **Phases filter**

Select the data pump phase for which you want to add a transformation. This advanced option is only available if the **Show multi-phase pump in DTS Designer** check box is selected in the **Package Properties** dialog box. The default is **Row transform** phase.

#### Name

View the name of the selected transformation. You must click on a mapping line to display the name of the transformation; otherwise **<none selected>** is displayed. By default, the transformations are named

**DTS\_Transformation\_***n*, where *n* is equal to the ordinal position of the mapped transformation.

#### Туре

View the type of column-level transformation for the selected mapping. You must click on a mapping line to display the type of the transformation; otherwise **<none selected>** is displayed. If you intend to create a new transformation, the value is ignored.

#### New

Display the Create New Transform dialog box, where you can select a type
of column-level transformation.

## Edit

Display the **Transformation Options** dialog box, where you can modify an existing transformation. Select the mapping line of the transformation you want to modify before clicking **Edit**.

## Delete

Delete a selected transformation. Select the mapping line representing the transformation you want to remove from the task before clicking **Delete**.

### Source

View the columns of the source data in graphical form, and click one or more columns to include in a transformation.

## Destination

View the columns of the destination data in graphical form, and click one or more columns to include in a transformation.

### Test

Run the selected transformation. Select the mapping line representing the transformation you want to run before clicking **Test**. The results of the transformation are sent to a text file, and you can view the data. No permanent changes to the destination data are made when you test a transformation.

## Select All

Select all source columns and all destination columns. Use **Select All** when you want to create a many-to-many type of transformation mapping.

## **Delete** All

Delete all transformations. Select the mapping line representing the transformation you want to remove from the task before clicking **Delete**.

## See Also

Mapping Column Transformations

Tasks That Transform Data

Transform Data Task

Multiphase Data Pump Functionality

# **Transform Data Task Properties (Lookups Tab)**

Use this tab to define a lookup query. Before defining a lookup query, you must first create connections for the source, binding, and lookup tables, and specify the source and binding tables.

## **Options**

#### Name

Type a name for the lookup.

## Connection

Choose an existing connection from the list on which to execute the lookup query.

## Cache

Type a value for or view the number of lookup results saved in a cache. Caching is especially useful with lookups if the number of rows being transformed is large, and you are querying on a small number of rows in the lookup table.

### Query

Build the lookup query in the Data Transformation Services (DTS) Query Designer. You can also paste the query from a text editor into the SQL pane.

### Add

Create a new lookup query, which will be listed as a row in the **Lookups** table.

### Delete

Removes the highlighted lookup query from the **Lookups** table.

## See Also

Lookup Queries

Configuring a Simple Lookup Query
Using ActiveX Scripts in DTS

# **Transform Data Task Properties (Options Tab)**

Use this tab to send rows to an exception file, determine the format of that file, and set the error count at which package execution should cease. The exception files are stored on a local or mapped drive. You can also specify items such as fast load, table locking properties, and constraint checking as data is moved.

## **Options**

#### **Exception file properties area**

#### Name

Specify the path and name of the file where exception records will be written. If the file does not exist at package run time, the file will be created. The file does not have a default extension assigned to it.

#### Browse

Display the **Select File** dialog box, where you can search the local computer or mapped drives for an existing exception file. If an existing exception file is used, the status and error information for the package will be appended.

### File type properties area

#### 7.0 format

Specify to save the exception file in 7.0 format. This format is useful if an exception file parser was written for Microsoft® SQL Server<sup>TM</sup> version 7.0 exception files, because exception files using this backwardcompatible format can still be used in the parser.

### **Error text**

Specify that any errors encountered during the task execution be recorded. Information such as the package name, execution start and completion times, and other data are entered in the exception log.

#### Source error rows

Specify that a separate exception file be created to contain all the rows

from the source data that did not get written to the destination. Formatting of the file is done according to the specified **File format properties**. The file name will be the same as the file name specified in the **Name** field, with the extension **.Source** appended to it.

#### **Dest error rows**

Specify that a separate exception file be created to contain records rejected from the source file. The file name will be the same as the file name specified in the **Name** field, with the extension **.Dest** appended to it.

## File format properties area

#### **Row delimiter**

Select the delimiter used to separate rows of data in the exception file. A carriage return/line feed **{CR}{LF}** is used by default.

#### **Column delimiter**

Select the delimiter used to separate the columns of data in the exception file. A vertical bar is used by default.

#### **Text qualifier**

Specify which character marks were used in the delimited data file to qualify text. Choose from: **Double Quote {''}**; **Single Quote {'}**; **<none>**. You can also type a character to use as the text qualifier.

#### Data movement properties area

#### Max error count

Set a limit for the number of errors allowed before processing is terminated for the task. When the SQL Server fast load option is selected, each error corresponds either to a row-level failure detected by the Transform Data task or to a batch failure. The value of **Max error count** includes the number of row-level errors detected by the Transform Data task plus batch failures. When the **Max error count** value is exceeded, task execution is terminated. The default is zero, which means that the task will terminate upon the first error.

#### Fetch buffer size

Set the number of rows of data being fetched at the source during data movement. Generally, you should not need to adjust this value unless it is necessary to optimize the characteristics of the data provider.

#### **First row**

Specify the first row of data to be moved. This is useful if the first row consists of column headings, or, if the first part of a data source has been copied. You can set this value to the row number where processing stopped in an earlier data pump operation.

#### Last row

Specify the last row of data to move.

## **SQL Server properties Area**

### Use fast load

Specify that you want to use high-speed bulk-copy processing. The fast load option can be used only when the destination connection is the Microsoft OLE DB Provider for SQL Server. When you enable this option, the data pump can accept batches of transformed data. Batch sizes are controlled through the **Insert batch size** option in this area.

## Keep NULL values

Specify that you want to keep the NULL value in the destination column, even if the destination table was created with a default value designated for the column. This option is available only if you enable **Use fast load**.

### **Check constraints**

Specify whether constraints on the destination table are checked during the load. By default, constraints are ignored. This improves the performance, but it also allows data that violates existing constraints to be inserted into the table. This option is available only if you enable **Use fast load**.

### Table lock

Specify how the table should be locked during use. When the **Use fast load** property is not used, and **Table lock** is not used, the table is locked

using row-level locks. If **Table lock** is used, the table is locked using table-level locking. If the **Use fast load** property is turned on, and **Table lock** is not used, the table is locked using row-level locks. If **Table lock** is used, then the table is locked using table-level locking. This option is available only if you enable **Use fast load**.

#### **Enable identity insert**

Allow explicit values to be inserted into the **identity** column of a table (SQL Server only). This option is available only if an **identity** column is detected. An **identity** column is defined as having a data type of **bigint**, **decimal**, **integer**, **numeric**, **smallint**, or **tinyint**, with the **Identity** property for the column set to **Yes**. This option is available only if you enable **Use fast load**.

#### Always commit final batch

Select to commit all rows in the final batch that were processed successfully before an error occurs. This property applies when a transformation or insert error occurs during processing of the final batch, so that all rows in the batch prior to the error do not have to be processed again. The setting is useful for large batch sizes. This option is available only if you enable **Use fast load**.

#### **Insert batch size**

Specify the number of rows in a batch. This option is available only if you enable **Use fast load**.

Values for **Batch size** work as follows:

If you set batch size to 0, the data is loaded in one batch, and the first row that fails will cause the entire load to be canceled and the step fails. This value is the default setting.

If you set batch size to 1, the data is loaded a single row at a time. Each row that fails is counted as a batch failure, and the value of Max error count is incremented by one. Previously loaded rows are either committed, or, if the step has joined the package transaction, retained in the transaction, subject to later commit or rollback. If you set batch size to a value greater than 1, the data is loaded one batch at a time. The first row that fails in a batch fails that entire batch; loading stops and the step fails. Rows in previously loaded batches are either committed, or, if the step has joined the package transaction, retained in the transaction, subject to later commit or rollback.

## See Also

Tasks That Transform Data

Transform Data Task

# **Transformation Flags**

Use this dialog box to customize data conversions between source and destination columns and enforce the stringency with which data type conversions are allowed. The transformation flags available through this dialog box correspond to a subset of the members of the **DTSTransformFlags** object class.

## **Options**

### **Default transformation flags**

Allow all possible data conversions. This is the least stringent data conversion setting. It corresponds to the **DTSTransformFlag\_Default** constant.

### Require exact match between source and destination

Enforce that data types match exactly in source and destination columns. If there is not an exact match, an error is generated. Use the **Advanced** tab of the **Data Transformation Properties** dialog box for error handling. This setting corresponds to the **DTSTransformFlag\_RequireExactType** constant.

#### **Custom transformation flags**

Enable selection of the three transformation flag settings listed in the following options.

#### Allow data type promotion

Allow data type promotion from 16-bit integer to 32-bit integer numbers. This setting corresponds to the **DTSTransformFlag\_AllowPromotion** constant.

### Allow data type demotion

Allow data type demotion from 32-bit integer to 16-bit integer numbers. This setting corresponds to the **DTSTransformFlag\_AllowDemotion** constant.

### Allow NULL conversion

Allow data conversions from columns that allow NULL values to columns that do not. This setting corresponds to the **DTSTransformFlag\_AllowNullChange** constant.

# See Also

**TransformFlags Property** 

# **Transformation Options (General Tab)**

Use this tab to name the transformation you want to create, view information about the transformation type, and display the Properties dialog box for the transformation.

When using other tabs in the **Transformation Options** dialog box, you must return to the **General** tab before you can view or configure the properties of the transformation by clicking **Properties**.

# Options

## Name

Type a name for the transformation, or use the default name. The default name is "DTSTransformation\_*n*", where *n* is equal to the number of the transformation being defined for the task.

## Туре

View the type of column transformation you selected in the **Create New Transformation** dialog box.

## **Properties**

Configure the properties of the selected transformation.

## Sequence

View the order in which the current transformation was created and will be executed. In transformations where the same destination column is written to by multiple transformations, the results of the last transformation (as designated by the sequence number) are saved.

## **Transform Description**

View a summary description of the transformation type selected.

## See Also

Mapping Column Transformations

Transformation Types Tasks That Transform Data

# **Transformation Options (Source Columns Tab)**

Use this tab to select and order the source columns you will use in the transformation. If you used the mapping lines on the **Transformations** tab of the Transform Data or Data Driven Query task to define the transformation, the source columns you mapped appear in the **Selected columns** list.

## **Options**

## Available columns

View the source columns from which you can select when creating the transformation.

### Selected columns

View the source columns that will be used in the transformation.

>

Copy a selected column from the **Available columns** list to the **Selected columns** list.

>>

Copy all columns from the **Available columns** list to the **Selected columns** list.

<

Remove a selected column from the **Selected columns** list.

<<

Remove all columns from the **Selected columns** list and from the transformation.

## See Also

Mapping Column Transformations

**Transformation Types** 

Tasks That Transform Data

# **Transformation Options (Destination Columns Tab)**

Use this tab to select and order available destination columns you will use in the transformation. If you used the mapping lines on the **Transformations** tab of the Transform Data or Data Driven Query task to define the transformation, the destination columns you mapped appear in the **Selected columns** list.

## **Options**

### Available columns

View the destination columns from which you can select when creating the transformation.

### Selected columns

View the destination columns that will be used in the transformation.

>

Copy a selected column from the **Available columns** list to the **Selected columns** list.

>>

Copy all columns from the **Available columns** list to the **Selected columns** list.

<

Remove a selected column from the **Selected columns** list.

>>

Remove all columns from the **Selected columns** list and from the transformation.

## See Also

Mapping Column Transformations

**Transformation Types** 

Tasks That Transform Data

# **Transformation Options (Phases Tab)**

Use this tab to select the transformation phase or phases you will implement by supplying and registering a COM object. You do not need to use this tab if you are supplying Microsoft® ActiveX® script functions to implement multiphase data pump functionality.

## **Options**

## Pre source data function

Call a function before the first fetch of source data. You create functions for the pre source transformation phase primarily for writing header rows to the destination.

## **Row transform function**

Call the default transformation function for copying and transforming data. Creating a function only for this transformation phase is the same as not using the multiphase data pump feature.

## **Post Row Transform Function**

Specify that post row transformation functions are executed after the row transform phase of the data pump. Only one of the following post row transform functions can be called for a row: **On insert success, On insert failure,** and **On transform failure**.

### **On insert success**

Create a function to be called on success of an Insert operation (or Insert query if the transformation is part of a Data Driven Query task). You cannot specify any destination operations in the returned status.

### **On insert failure**

Create a function to be called on failure of an Insert operation (or Insert query if the transformation is part of a Data Driven Query task). You cannot specify any destination operations in the returned status.

### **On transform failure**

Create a function to be called on failure of the normal transform phase (when the row transformation returns **DTSTransformStat\_Error** or **DTSTransformStat\_ExceptionRow**). Writing a function for this transform phase allows you to handle transformation errors (for example, type mismatches), overriding the value returned by the transformation and continuing with execution.

## **On batch complete**

Create a function to be called on success or failure of a batch or rows, as defined by the value specified in **Insert batch size** in the **Options** tab of the Transform Data task. Setting a batch size for a Data Driven Query task or parallel data pump task can only be done programmatically. Therefore, if you want to write an **On batch complete** function for either of those tasks, you should do so programmatically as well.

## On pump complete

Create a function to be called at the end of the transformation task (after all rows have been processed). Use functions written to **On pump complete** to free up resources and commit data held in global variables throughout the lifetime of data pump. You cannot access the data through an **On pump complete** function.

### Post source data function

Create a function for processing the destination data after completion of the task. Unlike **On pump complete** functions, functions written to this phase allow you to access the destination data. Common uses of a post source data function include writing footer rows to a file, freeing up resources, and committing data held in global variables.

## See Also

Multiphase Data Pump Functionality

Mapping Column Transformations

**Transformation Types** 

Tasks That Transform Data

# **Trim String Transformation Properties**

Use this dialog box to remove leading, trailing, and embedded white space from a string in the source column, and copy the result to the destination column.

## **Options**

## **Trimming Options**

Select the location in the string from which to remove white space.

## Trim leading white space

Remove any white space preceding the first character of the string data.

### Trim trailing white space

Remove any white space after the last character of the string data.

## Trim embedded white space

Remove any white space in between the first and last characters of the string data.

## **Case Options**

Select whether to change the case of the string data as part of the transformation.

### Do not change case

Leave the case of the string data intact after performing any trim operations.

## Uppercase

Change the case of the string data to uppercase after performing any trim operations.

### Lowercase

Change the case of the string data to lowercase after performing any trim operations.

# See Also

Trim String Transformation Middle of String Transformation

# **Unregister Custom Task**

Use this dialog box to remove any registered custom tasks, along with their registry entries.

## **Options**

## **Task Description**

Select the custom task you want to unregister from the list.

# See Also

DTS Custom Task Building a DTS Custom Task

# **Verifying Transformations**

Use this dialog box to select how you want the column mappings handled when the source or destination connection has changed and when column mappings exist that are no longer valid.

## **Options**

## **Remove invalid transformations**

Remove all column mappings. If you select this option, the **Transformations** tab is displayed, where you can reapply appropriate column mapping and transformations. By using this option to clear out all invalid mappings, you do not need to manually delete all invalid mappings and then set up appropriate column mappings. Thus, new mappings are more easily applied.

## **Change source/destination**

Modify the source or destination tables. If you select this option, the **Source** tab of the transformation task is displayed, where you can change the source table, or the **Destination** tab is displayed, where you can modify the destination.

### Remove all transformations and redo auto-mapping

Remove all the original column mappings and reapply the column mapping with the new fields. If you select this option, the **Transformations** tab is displayed, where you can review and apply appropriate column mappings and transformations. This selection maps every column automatically. Any incorrect mapping must be deleted and then reapplied appropriately.

## See Also

Mapping Column Transformations

# **View Data**

Use this dialog box to inspect source data that will be used by a transformation task, or to review the results of a test run of a transformation.

When used after testing a transformation, the **View Data** dialog box allows you to view the temporary text file containing the transformation results. You can check the data to ensure that the transformation is producing the expected results.

Because test mode sends data to a file rather than the actual destination, problems with the provider, the destination, or data type overflow may not be detected. Therefore, in certain cases, test results may not accurately represent the performance of a transformation task.
# **Workflow ActiveX Script Properties**

Use this dialog box to specify the scripting code that will perform the functions you need to customize a Data Transformation Services (DTS) package step (for example, retrying a connection).

# **Options**

#### Language Tab

This tab contains a list of the functions available in the language that you choose in the **Language** list.

#### Language

Select an available scripting language. When you install scripting languages on the computer, this list will update automatically. Microsoft® Visual Basic® Scripting Edition (VBScript) and Microsoft JScript® are available by default.

#### Functions

Select a function from the script language library to be placed into the **ActiveX script text box**. Double-clicking a function name inserts the function code into the text box at the position of the cursor.

#### **Entry Function**

Specify the name of the function that will be the entry point when the script runs. Only one function can be specified as an entry point for a workflow Microsoft ActiveX<sup>®</sup> script.

#### **Browser Tab**

View the constants available for use as return codes, package global variables, and lookups with a workflow ActiveX script. Click a tab to display the available information for selection, and double click on an item to copy the information into the **ActiveX script text box**.

#### ActiveX Script text box

Type or paste the scripting code necessary to perform the functions you need. The scripting editor has limited functionality; it does not include features such as statement completion or color-coding of reserved words. Comment lines use the apostrophe (') character syntax in any column, and all text from the comment character to the end of the line is ignored.

# Auto Gen.

Generate a single function placeholder in the selected scripting language. This option may not work for all installed scripting engines.

**IMPORTANT** If there is scripting code in the **ActiveX script text area**, the code will be deleted when you click **Auto Gen**.

## Browse

Display the **Select File** dialog box, where you can select a file containing code. When you select a file, this will copy the contents of that file into the **ActiveX script text box**.

**IMPORTANT** Opening an external script file from this dialog box will overwrite any existing code in the ActiveX script task text box. Therefore, this task should always be performed first.

## Parse

Check the syntax of the code for errors.

# Save

Display the **Save As** dialog box, where you can save all the code in the **ActiveX script text box** into a file on the local hard drive or on any mapped drive.

## Undo

Reverse a limited number of text entry actions. You cannot undo actions such as adding code through the **Select File** dialog box.

# See Also

Using ActiveX Scripts in a DTS Workflow

Using ActiveX Scripts in DTS Using Global Variables with DTS Packages Debugging ActiveX Scripts Using Return Codes in DTS DTS Designer Help

# **Workflow Properties (Precedence Tab)**

Use this tab to view and configure the relationships among tasks, transformations, and precedence constraints.

- If you are configuring the properties of a precedence constraint, this tab displays the steps, from source to destination, that must be completed for the constraint to be exercised. You can also use the tab to add additional constraints to the preceding steps, delete constraints on those steps, or redefine existing constraints for those steps.
- If you are configuring the workflow of a Transform Data task, this tab displays the source precedence constraints preceding the target transformation. You can also use the tab to add additional constraints to a workflow, delete constraints, or redefine existing constraints.

The order of constraints is not important. By default, each step is eligible for immediate execution.

# **Options**

#### Source step

Select an available source step from the list.

## Precedence

Specify task execution based on **On completion**, **On success**, or **On failure** status. You can also select **<none>** to terminate an existing relationship between a **Source Step** and **Destination Step**.

## **Destination step**

Select an available destination step from the list.

#### New

Add a new constraint between two steps, which you then need to configure.

#### Delete

Delete an existing constraint between two steps.

# See Also

DTS Package Workflow

DTS Designer Help

# **Workflow Properties (Options Tab)**

Use this tab to customize the workflow properties of a Transform Data task regarding transactional behavior, execution options, and the use of a Microsoft® ActiveX® script.

# **Options**

#### Description

View the name of the Transform Data task.

## Join transaction if present

If you select this option, and transactions are enabled, the step joins the Data Transformation Services (DTS) package transaction. Updates accumulate until commit or rollback. If cleared, updates are carried out one at a time, as they are requested.

#### Commit transaction on successful completion of this step

If you select this option, successful step completion triggers a transaction commit. Pending updates are made permanent. If cleared, any updates remain in the transaction until a later commit or rollback.

## **Rollback transaction on failure**

If you select this option, step failure triggers a rollback of the package transaction. Pending updates are discarded. If cleared, any updates remain in the transaction until a later commit or rollback.

## Execute on main package thread

Force the task associated with this step to execute on the main package thread rather than on a spawned thread. Choose this option if you are executing a task against a data provider that is not free-threaded and does not support parallel execution of tasks.

**IMPORTANT** If parallel execution is attempted on a provider that does not support it, serious errors may result. Some data providers used with DTS Designer that do not support parallel execution are the Microsoft OLE DB Provider for Jet, and the providers for Microsoft Excel, dBase, Paradox, and HTML source files. Use the **Execute on main package thread** option if more than one operation is used with these providers.

#### **Close connection on completion**

Close the connection after the completion of a package if the task is associated with a connection. Use of this option depends on the data provider; some providers perform better if the connection is not kept open. Another consideration on whether to select this option is the number of connections available for use and the expense associated with maintaining an open connection. For more information, see the OLE DB provider documentation.

## Fail package on step failure

Select this option to terminate processing of the DTS package if a failure occurs on this step.

#### **DSO rowset provider**

Expose OLE DB rowset data obtained from this step to an external consumer. This allows the package to be queried and for data from the step to be used as a source for other packages.

#### **Disable this step**

Disable this workflow step when the package is executed.

## **Step priority**

Set the Microsoft Windows NT® 4.0 thread priority for the step. The default setting is **Normal**.

## **Use ActiveX script**

Include an ActiveX script to execute the step.

## **Properties**

Display the **ActiveX Script Properties** dialog box, if **Use ActiveX Script** is selected.

# See Also

DTS Package Workflow

**Configuring Properties for DTS Transactions** 

Using ActiveX Scripts in a DTS Workflow

DTS Designer Help

# **Write File Transformation Properties**

Use this dialog box to copy the contents of a source column (data column) to a file whose path is specified by a second source column (file name column).

# **Options**

## Directory

Type the directory in which the files specified in the file name column are located or browse for the directory. At run time, each of the entries in the file name column is appended to this directory name to create a save path. If no file exists at that location, one is created and initialized with the contents of the data column.

## File type

Select whether the files to which the transformation writes data are **ANSI**, **Unicode**, or **OEM**.

#### File name column

Select the column containing the list of file names from the list.

#### Handle existing file

Select an option for situations where the file in which to write data already exists.

## **Overwrite if file exists**

Replace the contents of the existing file with new content (default selection).

## Append if file exists

Retain the existing content of the file and add the new content to the end of the file.

#### Error if file exists

Terminate processing and fail the write file transformation if the file exists.

See Also

Write File Transformation